ACORN USER

For BBC micro, Atom and Econet users

February 1983 £1

The Computer Programme - new TV series

Machine code answers

3D Atom graphics

1MHz bus details

Dynamic procedures

£80 printer reviewed



CONTENTS

ACORN USER

FEBRUARY 1983, NUMBER SEVEN

Editor
Tony Quinn
Editorial Assistant
Kitty Milne
Managing Editor
Jane Fransella
Production
Susie Home
Marketing Manager
Paul Thompson
Promotion Manager
Pat Bitton
Publisher
Stanley Malcolm
Designers and Typesetters
GMGraphics, Harrow Hill
Graphic Designer
Phil Kanssen
Printed in Great Britain
by E.T.Heron & Co. Ltd
Advertising Agents
Dealer Deals Ltd.
20 Orange Street
London WC2H 7ED
01-930 1612
Distributed to the News Trade
by Magnum Distribution Ltd.
72-8 Fleet Street,
London EC4Y 1HY.
Tel: 01-583 0961
Telex: 893340 Magnum G.
Published by
Addison-Wesley Publishers Ltd.
53 Bedford Square,
London WC1B 3DZ
Telephone: 01-631 1636
Telex: 881 1948
ISSN: 201-17002 7



Cover design by Chris Gilbert. Pictures from opening titles to BBC TV series – page 8

How to submit articles
You are welcome to submit articles to the Editor of *Acorn User* for publication. *Acorn User* cannot undertake to return them unless a stamped addressed envelope is enclosed. Articles should be typed or computer written. Black and white photographs or transparencies are also appreciated. If submitting programs please send a cassette or disc. Payment is £50 per page or pro rata. Please indicate if you have submitted your article elsewhere. Send articles, reviews and information to: The Editor, *Acorn User*, 53 Bedford Square, London WC1B 3DZ.

Coming soon in *Acorn User:*

● Software reviews ● Beeb hardware ideas ● Guide to printers ● Games listings ● BBC telesoftware launch ● Atom utility boards ● Programming in primary schools ● Music on the Beeb ● Language ROMs from Acornsoft ●

Subscription Information
For UK Subscriptions, send your cheque or postal order made payable to Addison-Wesley Publishers Ltd to:
*MAGSUB (Subscription Services) Ltd, Ground Floor Post Room
Oakfield House
Perrymount Road
Haywards Heath
West Sussex RH16 3DH*

Annual subscription rates

UK	*£15*
Europe	*£18*
Middle East	*£20*
The Americas and Africa	*£22*
Rest of the World	*£24*



THE RIGHTS AND WRONGS

WE DID it again last issue. We upset people at Acorn with the magazine. It happens every issue, and they have a moan and we have a moan back.

This time it was the programmers. They've gone through all the trouble of providing a useful compiler in the Beeb, and articles in *Acorn User* tell people to do it another way.

Their real beef is that this gets people into bad habits. If you play around with machine code and move things around in memory through any other method other than that described in the *User Guide*, you can mess things up when your machine is expanded.

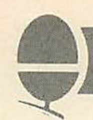
Hence, programs written with these 'bad habits' will not run once the Tube is fitted or you start using second processors. But, as always, the decision is with you, if you want to do things wrong, then you will do.

The same goes for programming techniques. In a perfect world no one would have a GOTO out of a PROC, but many people – especially those with no formal training – will do exactly this. And, after all, it is at these people that the BBC's Computer Literacy Project is aimed.

So in future we'll probably tell you the article doesn't do things 'right'. The program will run, but it is unlikely to work on machines which use the Tube or have the other expansions the Beeb is (or will be) famed for.

Another thing to remember is that many of the articles are written by users, not experts. Their techniques won't be perfect, but their ideas are good and worth being seen. The article may also just be a clue to something which needs developing.

Anyway, thanks for the feedback. If you don't like something say so – and suggest a better way if you can.



The artier side of Acornsoft

TRAVELLERS on London's Underground system might have been surprised to see Acornsoft's name as sponsors of an art exhibition.

The 34 wax oil pastel portraits on the theme of 'women in the news' replaced the normal advertising on the down escalator at Bank station.

Artist Victoria Preston specialises in tapestry portraits which, said Acornsoft director David Johnson-Davies, is a tech-nique allied to computer graphics. If you don't believe him, compare the front cover of September's *Acorn User* to a piece of tapestry.

Victoria, who has done portraits for President Hussein of Iraq and President Stevens of Sierre Leone, booked the advertising space herself for £800, and then sought sponsorship.

She saw it as taking art to the people, and a welcome change from the usual static way of presenting art at exhibitions.

In fact, a spokesman for London Transport estimated that up to a million people could have seen the exhibition in a month.

The portraits range from the Queen and Queen Mother, through Mrs De Lorean to a pensioner who was recently mugged. We've shown the 'nice' picture of Mrs Thatcher, but there was also one showing her darker side.

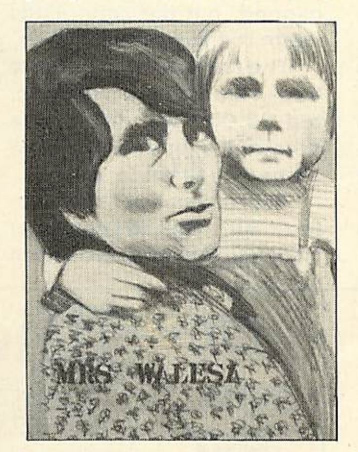
All of the portraits were based on news photographs of the people they depict. The exhibition was, unfortunately, due to end in January.

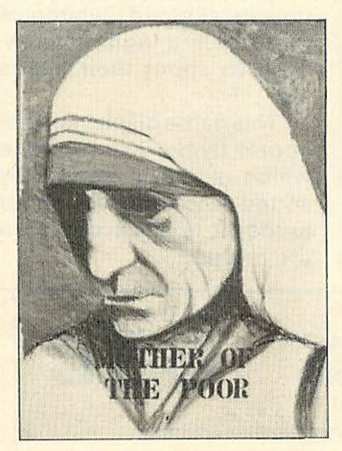
David Johnson-Davies was keeping quiet about the whole idea, but it appears there is a history in Acorn of sponsoring the arts.

Chris Curry, joint managing director of Acorn, once backed a harpsicord player at the Wigmore Hall. Rumour also has it that a Cambridge choir has approached him for financial support.

The head of London Transport's advertising department Joe Putnam is always ready to look at new ideas for the Tubes. So if you think your graphics might qualify as computer-generated art, give him a ring.









Four views of women: the nice side of Mrs Thatcher, Mrs Walesa, Mother Theresa and Billy Jean King

Courses and addresses

☐ Artificial Intelligence and Education is the theme of a conference to be held at Exeter University on April 16–17.

Seymour Papert (as in Logo) will be there and tutorials will be held on Prolog and Logo.

Details from Masoud Yazdani, Computer Science, Exeter University, Physics Building, Stocker Rd, Exeter EX4 4QL. Tel: (0392) 77911 ext. 216.

☐ Bristol University is running several courses using BBC micros. Contact D. Wilde or Mrs. L. Skinner, Extramural studies, 32 Tyndalls Park Rd, Bristol BS8 1HR. Tel: (0272) 24161

Be a computer sport

THE editors of a book on microcomputers in sport are looking for specimen programs which demonstrate the use of computers in sport, recreation and physical education. If any readers have written, or are prepared to write suitable progams and would like to submit them for publication, write as soon as possible to: Dr David Brodie, School of Physical Education and Recreation, University of Liverpool, PO Box 147, Liverpool L69 3BX.

The editors are interested in both simple and complex programs. Every one selected for inclusion in the book will have full author acknowledgement in the text and frontispiece.

Dr Brodie recently set up a sports injuries clinic which is used by students at Liverpool University. His department is a leader in the field of 'sports medicine', and the treatment of soft tissue injuries. Hugh Jones, winner of last year's London Marathon was a recent patient.

In honour of Gulnik

THE computer business spawns a new addition to our society's slang just about every week.

After vidiot (a video game junkie), a new name has surfaced, this time from the Acorn waterworks. It's gulnik, and is an affection term for computer show groupies.

It is mouthed in honour of a chap who is always popping up wherever Acorn exhibitions are to be found. Fame at last!

● December's *Acorn User* missed out the address of Bridge Software who make *Bridgeman*. The cassette costs £7.90 from 36 Fernwood, Marple Bridge, Stockport SK6 5BE. That'll teach them to put their address on cassettes!

Clive pays a festive visit to Acorn

ACORN and Sinclair seem to have buried the hatchet.

Any bad feeling between the two companies was undoubtedly played up by the computer press, but the appearance of Clive Sinclair at Acorn's Christmas party shows relations are improving.

Clive turned up with a small entourage which included his daughter. He wasn't particularly impressed with the atire of some of Acorn's programmers (casual, is how they see themselves) but apart from that the evening went well.

Meanwhile, the suggestion that Clive's brother is designing a new range of Acorn computers, has been described as exaggerated.

See next month's *Acorn User*, for more inside information from Cherry Hinton.



Fun at noisy northern show

by Barry Pickles

No, this is not an old picture of the Beatles, but three lads at the Northern Computer Fair playing with a piece of software.

The Helibomb package from Blackboard Electronics gave them a legitimate excuse to shout their heads off.

This game displays a helicopter flying over a bridge which is kept in the air by sound. And when you shout louder it drops a bomb on the bridge.

Helibomb was originally developed for the handicapped, but the schoolkids seem to have taken it over.

The fair was held at Belle Vue in Manchester. It attracted 23,000 visitors, and a permanent queue had developed by the final day.

Acorn was not represented, but the Manchester User Group and Northern Computers – who brought the Acorn caravan running Beebs under Econet – were there. Leasalink Viewdata also set up a stand.

Lynx and Dragon put on grand displays, but this didn't prevent the caravan being mobbed.

As usual at computer shows, the air conditioning could only put up token resistance to the crowds, so even in November, copious volumes of ice lollies were devoured.

WATFORD ELECTRONICS

DEPT BBC, CARDIFF ROAD, WATFORD, Herts, England
Tel Watford (0923) 40588. Telex: 8956095

MAIL ORDER AND RETAIL SHOP. TRADE AND EXPORT INQUIRIES WELCOME. GOVERNMENT AND EDUCATIONAL ESTABLISHMENTS OFFICIAL ORDERS ACCEPTED. Carriage: unless stated otherwise, please add min. 50p to all cash orders). VAT APPLICABLE TO UK CUSTOMERS ONLY. ALL PRICES EXCLUSIVE OF VAT. PLEASE ADD 15% VAT TO THE TOTAL COST INCLUDING POSTAGE. SHOP HOURS: 9.00am TO 5.00pm MONDAY TO SATURDAY. AMPLE FREE CAR PARKING. ACCESS ORDERS: Simply telephone through your order on Watford 50234/40589.

BBC MICROCOMPUTER Model B £399

(carr £7)

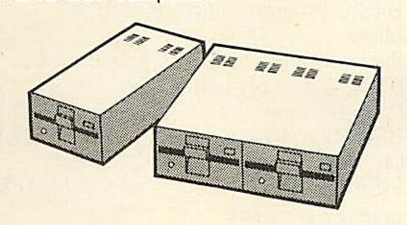
UPGRADE KITS. Upgrade your Model A to Mod. B with our Upgrade Kits and save yourself £ s s s ......

- BBC1 16K Memory (8 × 4816AP-3 100nS) £18.00
- BBC2 Printer User I/O Port £7.20
- BBC3 Disk Interface Kit £35.00
- BBC4 Analogue I/O Kit £6.65
- BBC5 Serial I/O Kit £6.35
- BBC6 Expansion Bus Kit £6.45
- Printer Cable Ready made 36" £11.95
- Complete Upgrade Kit Mod. A to Mod. B £43.65

Complete range of Connectors & Cables available for BBC Micro. Send SAE for list.

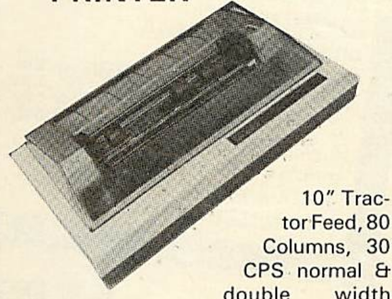
DISC DRIVES

'TEAC' BBC Compatible



- FD50A—Uncased, Single sided, 40 track, 5¼", 100K £125
- FD50A—Cased, Single sided, 40 track, 5¼", 100K £150
- Power Cable for above Discs £12
- FD50A—Cased with own Power Supply, S/S 40 track, 5¼", 100K £180
- FD50A—Twin Cased with own PSU, Single sided, 40 track, 5¼", 200K £350
- FD50E—Single cased with own PSU, Single sided, 80 track, 5¼", 400K £475
- FD50F—Twin cased with own PSU, Double sided, 80 track, 5¼", 800K £725
- Single Drive Cable for BBC Micro £8
- Twin Drive Cable for BBC Micro £12
- 10 Verbatim Diskettes, 5¼", S/sided £18
- 10 Verbatim Diskettes, 5¼", D/Sided £30

ACORN AP100A PRINTER



10" Tractor Feed, 80 Columns, 30 CPS normal & double width Characters. Dot res graphics Parallel Interface Standard.
Our price includes *FREE* 500 SHEETS of PAPER
Only **£175** (£7 carr)

- **SEIKOSHA GP250X** 10" Tractor Feed, 80 col. 50 CPS, normal & double width & height characters, RS232 & Centronics Interfaces standard. **£235** (£7 carr)
- **PRINTER CABLE** to Interface above printers to BBC Micro **£11**

NEC PC8023BE-C



100 CPS, Bi-directional, logic seeking, 80 columns, 7 × 9 Dot matrix, superscript & subscript, hi-res block graphics, underlining, true descenders, Tractor/Friction feed, Reverse linefeed, 2K Buffer, proportional spacing, attractively finished **£320** (£7 carr)

Interface Cable for BBC Micro **£12**

EPSON



MX80FT/3

10" Tractor & Friction feed, 9 × 9 matrix, 80 col., speed 80 cps, Bi-directional, hi-res bit image graphics. Subscript & superscript, Italics & underlining. Only **£324** (£7 carr)

Printer Interface Cable for BBC **£12**

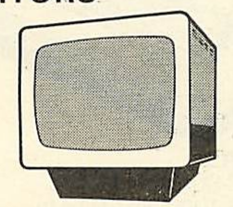
MX100FT/3

15" Carriage, 136 columns, plus all the facilities of MX80FT/3 Only **£425** (£7 carr)

LISTING PAPER

8½" or 9½" Fanfold paper plain or ruled (1000 sheets) **£7** (150p carr)
15" fanfold paper (1000 sheets) **£9** (150p carr)
Teleprinter Roll (econo paper) **£3** (150p carr)

MONITORS



MICROVITEC 1431. 14" Colour Monitor. RGB Input. (as used in BBC Prog.) Price including connecting Lead **£249.95** (carr £7)

ZENITH 12" Green Monitor. Excellent res. **£80** (carr £7)

CASSETTE RECORDER & ACC.

Top quality Sanyo Slim-line, portable Cassette Recorder. Ideal for Computer use. Mains/Battery operated. **£24** (carr £1)
C12 Computer Grade Cassettes in library cases **40p**

Cassette Interface LEAD for BBC

DIN Plug to 3 JACK Plugs **£2**
DIN Plug to DIN + JACK Plug **£2**

BBC SOFTWARE

(N.B. All Software is copyrighted)

ASTEROID BELT: (32K) Arcade type Space game. Destroy the cloud of Asteroid surrounding you with your photon belts **£7.95**

BEEBTREN: (16/32K) A realtime advanced Startrek Game. Facilities incl. 'Probe Satellite', 'damage report' & 'On board Computer'. **£6.95**

CHARACTERS: (16/32K) Makes redefining of Invaders, Foreign Characters, technical symbols etc, character shapes simpler. Clumsy Binary & Hexadecimal notations are not required any more. **£5.80**

CAR WARS/Alien PLANETS: (32K) Two graphics game. Excellent graphics, both Basic & Machine Code. **£6.25**

CUBE. A full colour stunning 3-D (colour graphics) of Rubik cube. Solve the jumble. **£7.80**

CHESS: (32K) Machine Code. An excellent game of enormous range of skill levels with wide variations. Uses high resolution colour graphics. **£9.95**

DISASSEMBLER: (16/32) Machine code. Ideal for discovering workings of MOS & BASIC. Invaluable for Ass. Lang. Programmer. **£5.80**

GALACTIC INTRUDER: (32K) Another arcade style game with hi-res colour & sound effects (Machine code) **£6.25**

HITCH-HIKER: (32K) An adventure game based on "Hitch-hiker's guide to the Galaxy". A fascinating game that tests your skills & wits. **£5.85**

MASTER MIND: 16/32K Mastermind game with a difference. Gives more fun (& frustration) than the original version. **£5.80**

POLAR BEAR: (32K) Educational game for 8-12 years old. Incorporates simple maths tables. **£6.50**

SPACE KINGDOM: (32K) Travel round the universe capturing and colonising Star systems to become the galactic ruler. **£6.95**

SNAKE: (32K) An arcade type game. Gives hours of fun. One of the best games available for this machine. Try it for yourself. **£7.80**

SOUND IDEA: (16/32K) A utility program for those interested in the sound and envelope facilities of BBC Micro. **£5.80**

UTILITY PACK 1: (32K) Envelope program/Variable print/Text scanner/Hex dump. **£6.75**

UTILITY PACK 2: (32K) Disassembler/Program, Crunch/Text Editor. **£6.75**

ZOMBIE'S ISLAND: (32K) Fight for your survival on this dangerous island inhabited by hungry and ferocious cannibals. (Basic & Machine code program) **£6.95**

BOOKS *(N.B. No VAT on Books)*

30 hour basic for BBC	£5.50
6502 Assembly Lang. Programming	£12.50
BBC Micro revealed	£8.00
Instant BBC Micro Machine Code	£5.00
Let your BBC Micro teach you to program	£6.50
Practical Programs for BBC & Atom	£6.90

WEROM©: ATOM Utility ROM. Watford's own most sophisticated but easy to use 4K ROM based on BASIC extension for Acorn ATOM. Plugs straight into the utility socket in an ATOM with floating point. The special features are: High Speed Tape Interface—Memory Dump, Modify—Machine Code Breakpoints—BASIC Error Trapping—READ, DATA & RESTORE—Full BASIC Keyboard Scanner (BBC like)—Find line, & Delete—Two-key Rollover Keyboard—FULL Disassembler—AUTO Line Numbering—PLUS: Chain, Cursor Movement, Loop Aborting—Easily extendible further. Supplied complete with instructions Only **£9.95**

SOFTWARE FOR ALL

PROGRAMS FOR BBC MICRO

SPACE KINGDOM

Your mission is to become Galactic Ruler. Explore neighbouring systems with your Scouts. Evaluate your chances if you attack. Prepare your fleet for the final assault. The more Systems you rule, the more powerful you become to finally rule the universe.

32K. £7.95.



POLAR BEAR



Educational game for 8-12-year-olds incorp. simple maths tables.

32K **£7.45**

ROW OF FOUR

Try to outwit your computer opponent in this game of skill. Great graphics. BASIC and machine code. 32K

£6.45

GALACTIC INTRUDER



Fast arcade-type action game with Hi-Res colour and sound. Machine Code. 32K **£6.95**

DATABASE for 16K/32K



Accepts up to ten definable fields. Facilities include quick search, sort and hard copy — capable of storing up to 300 complete records in memory. (Model B)

£9.95

TIME TRAVELLER



Travel through over 100 rooms in different times, and find the key back to your own time. 32K

£6.95

Utility Pack I **£7.45**

Envelope program, variable print-text scanner and memory dump. 32K

CAR WARS/ALIEN PLANET

Two addictive graphics games both BASIC and Machine Code.

32K **£6.95**

J.R.



Two player game, features include exploration, drilling, employment and Price Wars. 32K

£6.95

Utility Pack 2 **£7.45**

Disassembler, program crunch and text editor. 32K

ZOMBIE ISLAND



Fight for survival on an island inhabited by hungry, dangerous cannibals. BASIC and machine code program.



32K **£7.95**

BEEBTREK

Real time advanced Startrek Game. Extra facilities include "probe satellites", "damage reports" & "on-board computer".

16/32K **£7.95**

SEND TODAY! Post to SOFTWARE FOR ALL, Dept A 72 North Street, Romford, Essex or Phone: (0708) 60725.

DEALER ENQUIRIES WELCOME

Name: ....

Address: ....

....

☐ Please send me full details. I enclose S.A.E.

Please send me:

....	£ ....
....	£ ....
....	£ ....
....	£ ....
....	£ ....
....	£ ....
Please add £1.00 P&P per order..	£ 1.00
Total:	

I enclose Cheque/P.O. made payable to SOFTWARE FOR ALL.

Please bill my Access/Barclaycard/Amex No: ....

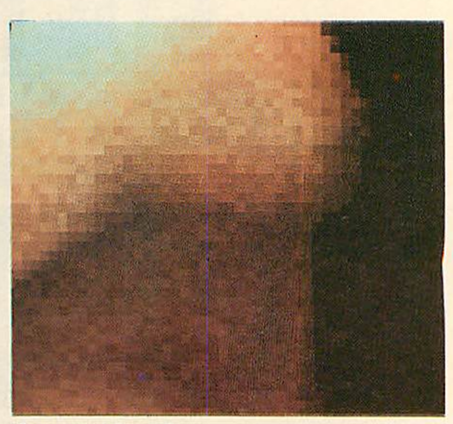


"Programs for the people"









These pictures show close-ups of the nose of the whole phantom head. Notice how the screen pixels become visible as the camera zooms in.

The Phantom Head

THESE pictures – shot for the BBC *Computer Programme* – show an eerie side to computing. This 'phantom head' has been digitised for use by orthodontic surgeons.

The three dimensional shape is stored in a computer's memory, and can be viewed and rotated at will. Surgeons at London's University College Hospital will use the image to plan operations.

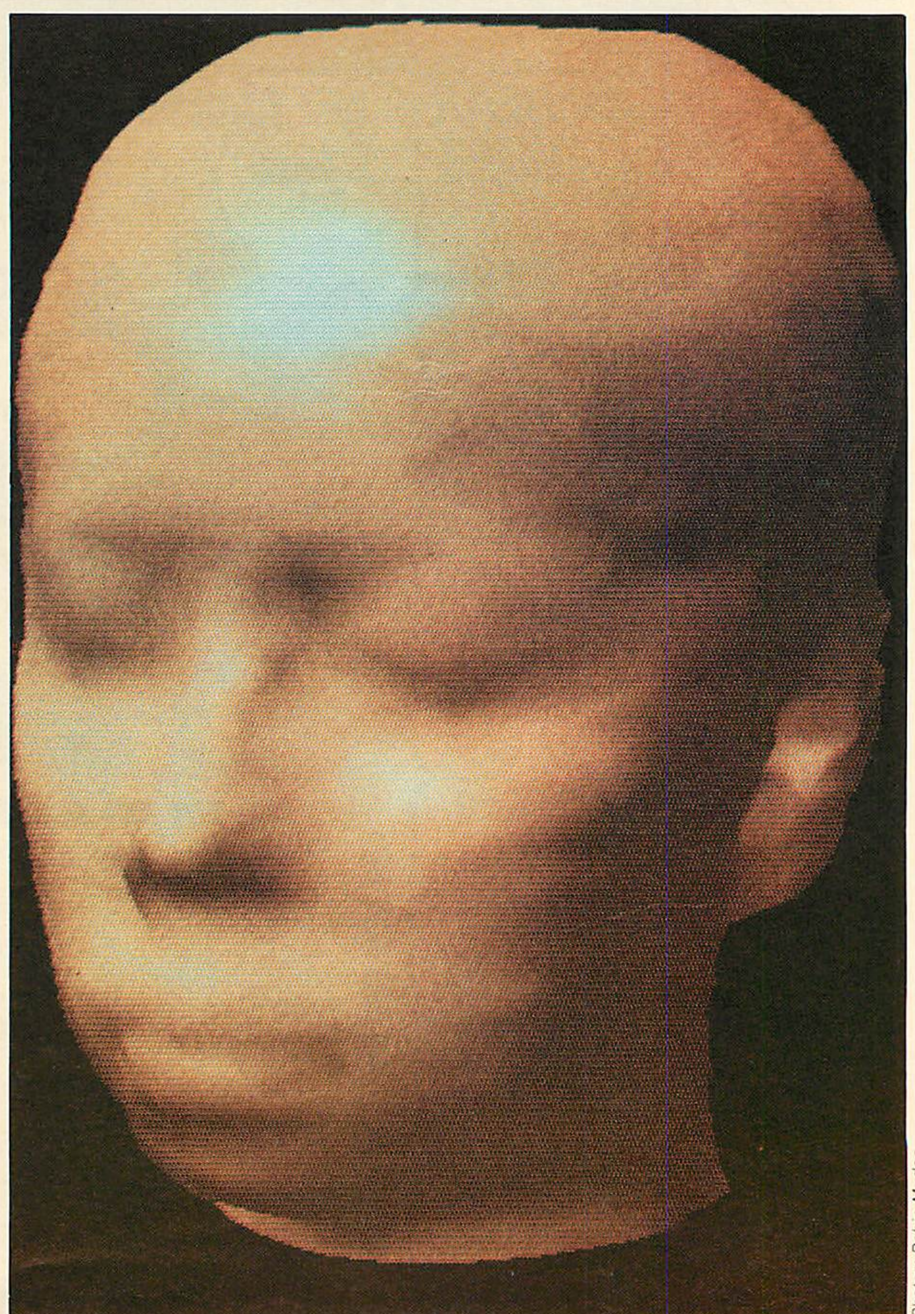
More details on this will be broadcast in the programme dealing with graphics on January 31.

This month's front cover shows Marvin the paranoid android (from *Hitchhiker's Guide to the Galaxy*) with Ian McNaught Davis, presenter of the TV series. Marvin appears in programme 2 which will be repeated on Sunday, January 24th.

The cover also shows some of the graphics used in the opening titles of the series. The owl image is formed of a series of rings, and was shown as a close-up in January's *Acorn User*.

Programmes are shown on Monday nights from January 10 on BBC1. Repeats are on Sundays at 12.35 from January 16. School transmissions are also on Mondays at 15.05, on BBC2.

You should see a lot more BBC micros knocking about – and they actually work, unlike the first series last year which often used mock-ups.



Photos: Robin Mudge

Computer Literacy One Year On

Where does the BBC go from here? Results of surveys commissioned during the first year of the Computer Literacy Project are now in and have been analysed. John Radcliffe, executive producer of the project, looks back on the year and outlines some future plans.

Early television programmes about information technology, such as *The Silicon Factor* which David Allen produced in 1979, were largely to do with the social impact of computers. The series examined questions which were very much in people's minds at that time: what effect will this technology have on my job? How will it affect my office?

Programmes about these issues met with great public response, more positive than on most technical subjects. But something else also happened. People started to ask us other things, basic questions about the technology.

These questions, coming from many quarters, at a time when personal computers were suddenly becoming much cheaper, led us to the computer literacy project.

The BBC decided the project would be built around a 10-part television series which would set out to explain some of the fundamental principles of computing in terms which would be accessible to a wide audience. Paul Kriwaczek, the producer of this first series, made a test programme, which we tried out on groups of possible viewers, and the BBC's broadcasting research department also asked 4,000 people about their attitudes towards computing.

This research discovered some surprising things. First, interest in the technology did not just extend to enthusiasts – the people who buy computer magazines or join computer clubs, or who use computers in their work.

We had expected the audience to consist largely of men, in the upper socio-economic categories, and aged between 20 and their middle 30s. In fact we found that interest extended to all sorts of people. And they didn't just want to be told in a general way about the technology – they wanted to learn about it. They seemed to have a greater propensity to learn about this than any other subject we had researched. But although people were curious, they were also anxious about their capacity to understand the technology.

Interestingly, there was an age element to this. Each age group thought the one below them would be better at the subject. The 30 year olds thought 20 year olds would do better: they were terrified of the teenagers and so on. This was clearly nonsense, but it was a powerful and pervasive myth.

So we set out to make a television series which would nail the myth, and convince people that computing was accessible to anyone prepared to make the effort to understand it.

The Computer Programme was transmitted three times in the earlier part of 1982, and twice again in the autumn. The BBC Broadcasting Research Department carried out an extensive survey of the impact of the series, and this seems to indicate that we have hit our target. Many people who were already interested in computing watched the series, and some of them found it too simple. But a great many others, who would probably have defined themselves as computer illiterate, also watched some of the programmes at least. Over half the audience were over 35, a quarter of them were over 45, and nearly half were women. And as many as seven million people saw one or more of the programmes.

Even more interesting was the response to other parts of the project. When we published David Allen's *The Computer Book*, we expected to sell 15-20,000 copies. In fact over 60,000 copies have been sold in the past year. When the National Extension College produced the correspondence course '30-Hour Basic', the estimated demand was 10-15,000 – in fact 100,000 copies of the course have been sold since January 1982.

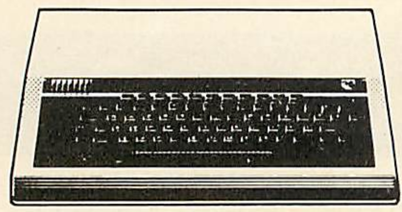
Leslie Morphy's computer referral service, which exists to put enquirers in touch with information about any aspect of the project, letters are coming in at a rate of 200 a week. When asked to estimate the demand for the BBC microcomputer, we estimated a need for up to 12,000 machines by the middle of 1982. In fact this many had been ordered before the series was launched.

This has encouraged the BBC to commit substantial resources to making more broadcast series about computing, to develop the capacity to produce software, to commit itself to the development of the new broadcast telesoftware service, and to lay plans for more correspondence courses.

We hope the second series of *The Computer Programme: Making the Most of the Micro*, will go some way towards satisfying the desire of many users for more advanced explanation. Plans are also being laid for further series in 1983/4 including one on 'The electronic office' another on 'Computers for control' and a magazine programme for computer users.

We hope that people who have comments or criticisms or suggestions won't hesitate to write and tell us about them.

Official BBC Dealer



BBC MODEL B £399
(price include VAT, carr. extra £8)
Complete Upgrade Kit £50
Installation £15
Individual Components also available
All mating connectors with cables in stock
WIDE RANGE OF SOFTWARE HELD IN STOCK
SEND or PHONE FOR OUR BBC LEAFLET

'VIEW' BBC Word Processor ROM	£ 52
Teletext Adaptor — — —	£ 196
Prestel Adaptor — — —	£ 90
2nd Processor (6502) + 64K RAM	£ 170
2nd Processor (Z80) + 64K RAM	£ 170

Please Phone to Check Delivery Details on New Addons

BBC DISC SYSTEM
Disc Interface inc. 1.0 operating System £95 Installation £20
BBC Single Drive (look) £230 + £6 Carr.
BBC Dual Drive (800K) £POA
A wide range of Software from Acorn, BBC and others for education and fun in stock.

BOOKS
Basic Programming on BBC £5.95
30 HR Basic (NEC) £5.95
Let your BBC teach you to Program £6.95
BBC Micro Revealed £7.95
Assy. Lang. Program on BBC £8.95
Program the 6502 £10.75 6502 Games £10.75
6502 Software Design £10.50
(No VAT on Books - £1 p&p/book)
For Full Details of Add-ons for BBC Send for our Leaflet.



PRINTERS

NEC PC8023 BEC
*80 Cols. 100CPS
*Proportional Spacing
*Hi-Res & Block Graphics
*Bi-directional Logic Seeking
*Forward & Reverse Line Feed
*International & Greek Alphabet
*Auto underline, Super & Sub Scripts
*2K Built in buffer * **£320 + £8 Carr**

Parallel Printer Lead **£13.50**
2000 Sheets 9 x 11½" paper **£13.50 + £3.50** p&p.

EPSON MX80 & 100F/T3
*MX80: 80 Cols. 80 CPS
*MX100:136 Cols. 100 CPS
*Bit Image Printing!
*Hi-Res Graphics
*Bi Directional Logic Seeking
*International Characters
*32 Print FONTS
*Auto underline
*Super & Sub Scripts
MX80 F/T3 **£325 + £8 Carr.**
MX100 F/T **£425 + £10 Carr.**

SEIKOSHA GP100A
*80 Cols. 30 CPS
*Self Testing
*Hi-Res Graphics
*Standard & Double width characters
*only for **£185 + £6 Carr.**

BBC COMPATIBLE 5¼" DISC DRIVES
These are TEAC mechanism fully compatible with BBC. They are supplied with independent power supply and housed in BBC matching cabinet.

Single Drives:	100K **£190**	200K **£255**	400K **£345**
Dual Drive:	200K **£360**	400K **£480**	800K **£610**

Carr. £6/Single drive £8 Dual drive. Disc Cable: Single £8 Dual £12

DISKETTES
in packs of 10
Single Sided 40 tracks **£15**
Single Sided 80 tracks **£24**
Double Sided 80 tracks **£32**
p& p £2/pack

MONITORS

Microvitec 1431 14" Colour Monitor **£269 + £8 Carr.**
Microvitec 2031 20" Colour Monitor **£389 + £8 Carr.**

Kaga 12" Colour Monitor RGB **£250 + £8 Carr.**
Kaga 12" Antiglare Green Monitor **£107 + £6 Carr.**
Hi-Res 12" Green Screen Antiglare Monitor **£99 + £6 Carr.**

Ferguson Cassette Recorder **£26 + £1.50 Carr.**
Cassette Leads
7pin DIN 3 jacks **£3.50**
7pin DIN-pin DIN + jack **£4.00**

We carry a wide range of connectors and assemblies, Microprocessors, RAMs, EPROMs, Crystals, etc. Price Lists, Leaflets available on request. Large stocks enable same day despatch on most orders. Special pricing for dealers pruchasing in quantity. Ask for dealer price list. Orders from schools, colleges and educational establishments welcome. For UK orders please add 40p p&p unless stated otherwise and VAT at 15% to the order value. No VAT on overseas orders. P&P on application.

TECHNOMATIC LTD.

MAIL ORDERS TO: 17 BURNLEY ROAD, LONDON NW10 1ED
Tel. 01-452 1500/450 6597 Telex 922800
RETAIL SHOPS: 15 BURNLEY ROAD, LONDON NW10 305 EDGWARE ROAD, LONDON W2



Finding a home for machine code

Once outside the protection of Basic, using memory is a minefield for the unwary. John Ferguson and Tony Shaw act as your guides

Close examination of the Beeb's memory map reveals many contenders jostling for space within the computer's random access memory (*User Guide,* p500). In the model B, 32k bytes of RAM are used not only for Basic programs but also for:

- the memory mapped screen; up to 20k can be devoted to the screen.
- operating system functions, eg, soft key buffer, input buffer, 6502 stack.
- storage of dynamic variables.
- the Basic stack, a 'notepad' used to hold return addresses from GOSUBs and PROCedures.

In fact, RAM is a minefield for anyone who steps outside the protection of Basic. Hence, care must be taken when choosing a position for a machine code program if corruption of either Basic or the machine operating system is to be avoided.

To keep clear of possible disaster, several strategies can be followed. The first is based on having a detailed knowledge of the memory map and knowing where to position machine code safely. Program 1 illustrates the technique. The P% variable is given the value of &1500, a known starting point of some safe memory locations well separated from Basic, the machine operating system and the screen memory.

The program prints the message 'FRED' on the screen, making use of the $ or 'string indirection operator' *(User Guide* p409) to place the characters for 'FRED' plus a carriage return in consecutive memory locations, starting at MESSAGE (&1200). Again, memory locations &1200 through to &1204 were chosen as safe locations.

This is satisfactory when working with relatively small programs. If, however, extra lines are added to the Basic program there is a danger that it can 'creep' up the memory map and eventually overlap the area set aside to hold both the message and the machine code.

A second approach, useful for machine code programs up to 256 bytes long, is to locate the code in the memory slot (&0D00 to &0DFF), called page D. This area, just below Basic, has been set aside specially for 'user supplied resident routines'.

Program 2 places machine code in this region. A different technique is used to locate the characters for

page 14 ▶

```
   10 REM SHORT PROGRAM TO PRINT "FRED"
   20 OSASCI=&FFE3
   30 REM PLACE CHARACTERS FOR "FRED"
   40 REM FROM &1200 UPWARDS
   50 MESSAGE=&1200
   60 REM INSERT "FRED"
   70 $MESSAGE="FRED"
   80 P%=&1500
   90[
  100 .START LDA MESSAGE ;GET FIRST BYTE
  110        JSR OSASCI  ;ONTO SCREEN
  120        LDA MESSAGE+1 ;GET SECOND BYTE
  130        JSR OSASCI ;ONTO SCREEN
  140        LDA MESSAGE+2 ;ETC FOR REMAINING
  150        JSR OSASCI ;TWO BYTES
  160       LDA MESSAGE+3
  170       JSR OSASCI
  180        RTS          ;BACK TO BASIC
  190]
  200 CALL START
  210 END
RUN
1500
1500 AD 00 12 .START LDA MESSAGE ;GET FIRST BYTE
1503 20 E3 FF JSR OSASCI  ;ONTO SCREEN
1506 AD 01 12 LDA MESSAGE+1 ;GET SECOND BYTE
1509 20 E3 FF JSR OSASCI ;ONTO SCREEN
150C AD 02 12 LDA MESSAGE+2 ;ETC FOR REMAINING
150F 20 E3 FF JSR OSASCI ;TWO BYTES
1512 AD 03 12 LDA MESSAGE+3
1515 20 E3 FF JSR OSASCI
1518 60       RTS          ;BACK TO BASIC
FRED
```


Program (1)
Machine code and message in fixed locations between basic program and basic stack – (but beware!)



```
 10 REM SHORT PROGRAM TO PRINT "FRED"
 20 OSASCI=&FFE3
 50 FOR PASS =1 TO 3 STEP 2
 60 P%=&0D00
 70[OPT PASS
 80 .START LDA MESSAGE ;GET FIRST BYTE
 90        JSR OSASCI   ;ONTO SCREEN
100        LDA MESSAGE+1 ;GET SECOND BYTE
110        JSR OSASCI ;ONTO SCREEN
120        LDA MESSAGE+2 ;ETC FOR REMAINING
130        JSR OSASCI ;TWO BYTES
140      LDA MESSAGE+3
150      JSR OSASCI
160       RTS          ;BACK TO BASIC
170]
180 REM DEFINE START OF "FRED" MESSAGE
190 MESSAGE=P%
200 REM INSERT "FRED"
210 $MESSAGE="FRED"
220 NEXT PASS
230 CALL START
240 END

0D00 AD 19 0D .START LDA MESSAGE ;GET FIRST BYTE
0D03 20 E3 FF JSR OSASCI   ;ONTO SCREEN
0D06 AD 1A 0D LDA MESSAGE+1 ;GET SECOND BYTE
0D09 20 E3 FF JSR OSASCI ;ONTO SCREEN
0D0C AD 1B 0D LDA MESSAGE+2 ;ETC FOR REMAINING
0D0F 20 E3 FF JSR OSASCI ;TWO BYTES
0D12 AD 1C 0D LDA MESSAGE+3
0D15 20 E3 FF JSR OSASCI
0D18 60       RTS          ;BACK TO BASIC
FRED>
```

&8000
DEFAULT HIMEM
Basic stack
LOMEM
Basic Program
PAGE (&0E00)
Message
Start (&0D00)
Machine code
Operating system use
&0000

0D1D	0D	C-RTN
0D1C	44	"D"
0D1B	45	"E"
0D1A	52	"R"
0D19	46	"F"

RAM memory map (not to scale)

Program 2
Machine code and message located in 'Page D' – reserved space for user supplied resident routines

```
 10 MODE 7
 20 REM LEAVE 1 PAGE FREE ABOVE HIMEM
 30 REM (ON MODEL A SET HIMEM=&3B00)
 40 HIMEM=&7B00
 50 REM SHORT PROGRAM TO PRINT "FRED
 60 OSASCI=&FFE3
 70 FOR PASS =1 TO 3 STEP 2
 80 P%=HIMEM
 90[OPT PASS
100 .START LDA MESSAGE ;GET FIRST BYTE
110        JSR OSASCI   ;ONTO SCREEN
120        LDA MESSAGE+1 ;GET SECOND BYTE
130        JSR OSASCI ;ONTO SCREEN
140        LDA MESSAGE+2 ;ETC FOR REMAINING
150        JSR OSASCI ;TWO BYTES
160      LDA MESSAGE+3
170      JSR OSASCI
180       RTS          ;BACK TO BASIC
190]
200 REM DEFINE START OF "FRED" MESSAGE
210 MESSAGE=P%
220 REM INSERT "FRED"
230 $MESSAGE="FRED"
240 NEXT PASS
250 CALL START
260 END

7B00 AD 19 7B .START LDA MESSAGE ;GET FIRST BYTE
7B03 20 E3 FF JSR OSASCI   ;ONTO SCREEN
7B06 AD 1A 7B LDA MESSAGE+1 ;GET SECOND BYTE
7B09 20 E3 FF JSR OSASCI ;ONTO SCREEN
7B0C AD 1B 7B LDA MESSAGE+2 ;ETC FOR REMAINING
7B0F 20 E3 FF JSR OSASCI ;TWO BYTES
7B12 AD 1C 7B LDA MESSAGE+3
7B15 20 E3 FF JSR OSASCI
7B18 60       RTS          ;BACK TO BASIC
FRED>
```

&8000
Memory mapped screen
&7C00 in Mode 7, Model B
HIMEM (&7B00)
Machine code
Basic stack
LOMEM
Basic program
&0000

7B1D	0D	C-RTN
7B1C	44	"D"
7B1B	45	"E"
7B1A	52	"R"
7B19	46	"F"

RAM memory map (not to scale)

Program 3
Machine code and message located above defined value of HIMEM (Note – selection of another screen mode will wipe out machine code program).

BIGGEST AND BEST

We're best because we're biggest. The biggest Acorn Distributor you can get.
We've got the biggest range of hardware, software, add-ons, books, stationery, peripherals, monitors and printers. And the most efficient distribution facilities in the country.

Don't try the rest—come to the best!



230/6, Derby Road, Stapleford, Nottingham NG9 7BL Tel: Nottingham (0602) 396976



Figure 1. Addressing Modes

The effect of many of the microprocessor's instructions can be varied by making use of the many addressing modes available. The mode used will determine how the microprocessor will obtain data that is to be used with the instruction. However, the overall effect of the instruction will be the same whatever the mode. For example, LDA will always result in the accumulator being loaded with data no matter which is used. Exactly where the data will come from, however, may vary greatly as the following examples illustrate.

Addressing mode	Assembler statement	Hex code
Absolute	LDA &2034	AD 34 20
Zero page	LDA &46	A5 46
Immediate	LDA #&2C	A9 2C

In the absolute addressing mode example, the data loaded into the accumulator is that contained in address location &2034. When the Assembler translates the statement LDA &2034 it assigns the code AD to the LDA since it has determined that absolute addressing is intended. The address, in lo-byte/hi-byte order, then follows. (This placing of the address data in reverse order is for the benefit of the microprocessor – when it receives 16 bit addresses it requires the low order byte before it receives the high order byte.

Zero page addressing is a special case of absolute addressing that assumes address values will start with 00. Thus LDA &46 loads the accumulator with the data contained in location &0046 and two bytes are all that are necessary to store this instruction in machine code form.

The final example uses immediate (or direct) addressing. No address locations are used but instead the data to be loaded immediately follows the instruction – &2C in this case. The microprocessor then knows on receiving AD, A5 or A9 exactly where it must get the data to be loaded into the accumulator (figure 2.1).

These examples represent only three of the many addressing modes available to the 6502 programmer. One further mode worth a mention in this brief overview is implied addressing. This is the simplest and is used by instructions that have only one way in which they might be used. DEX (DEcrement the X register) and TYA (Transfer the Y register contents to the Accumulator) are both examples of instructions that use only implied addressing.

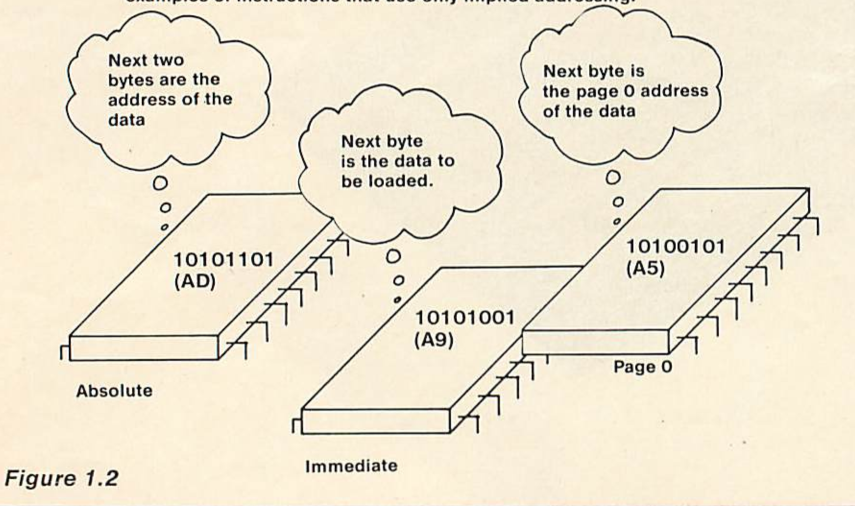


Figure 1.2

the message 'FRED'. MESSAGE is set equal to the value of the location counter at the end of the program, ie the address of the first available location. Characters for 'FRED' are then loaded into memory from this address upwards. It is worth noting that the program requires a two pass assembly to resolve a value for the forward reference MESSAGE.

A third technique can be used by those familiar with the system variable HIMEM (*User Guide* p270). All memory locations up to HIMEM are assumed by Basic to be available for use. The default value of HIMEM is just below the memory mapped screen. Below HIMEM, Basic uses some RAM locations as a notepad or stack, storing return addresses from GOSUBs and PROCedures.

However, HIMEM can be altered by the programmer to a new value further down the memory map, leaving enough space between the bottom of the screen memory and HIMEM to house the machine code. Program 3 illustrates this. Note how the new value of HIMEM is established at the start of the program, in case it is followed by any instructions that use the Basic stack.

Finally, program 4 uses a DIM statement (*User Guide* p236) to reserve space at the top of the Basic program for machine code. Line 40 sets aside 100 bytes, more than enough to hold both the machine code program and the characters for the message. Using this technique, Basic takes care of the position of the machine code. If extra lines are added to the Basic program, the position of the final code is 'shunted' up the memory map as the Basic program grows.

The 6502 microprocessor has many addressing modes which form an important part of the programmer's armoury (figure 1). Indexed addressing is particularly useful when processing the contents of a continuous area of memory, such as lists or tables of data.

The following statement illustrates how the LDA instruction uses indexed addressing

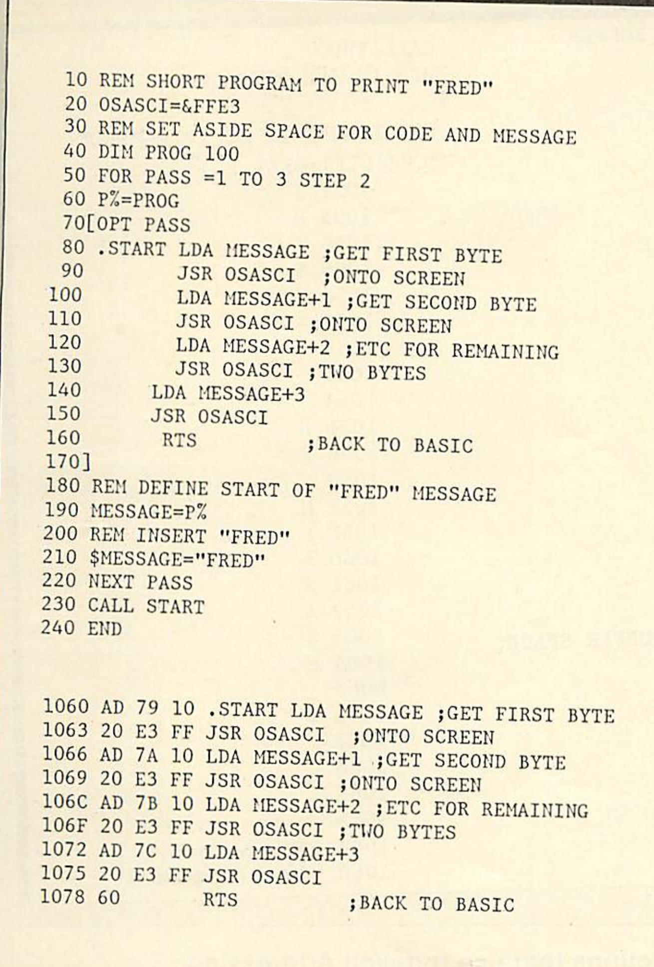
	Hex code
LDA &3025,X	BD 25 30

Here, as always, LDA loads the accumulator with data. The microprocessor determines an address from which to load the data by adding the address supplied (the base address) to the contents of the X register. If the X register contains &10, then, in the above example, the accumulator will be loaded with data from address &3035 since &3025 + &10 = &3035. As the X register is only eight bits in size, indexed addressing is restricted to a displacement of 255 bytes from the base address (maximum = base address + &FF0.

There is also zero page indexed addressing. Furthermore, either the X or Y register's contents might be used as the index to add to the base address. In total there are four such indexed addressing modes – absolute, X; absolute, Y; zero page, X, and zero page, Y. Of the 6502 instructions that can employ indexed addressing, not all can use all of these four modes. Table 1 lists the instructions that can be used in each of these four indexed addressing modes. Other modes make use of indexed addressing, but for now we will use the four mentioned.

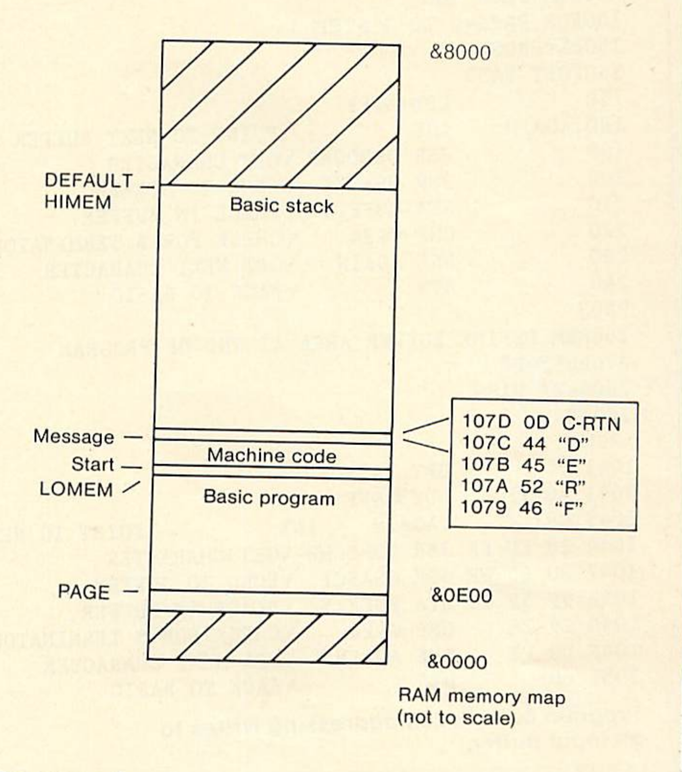
Program 5 illustrates how





```
10 REM SHORT PROGRAM TO PRINT "FRED"
20 OSASCI=&FFE3
30 REM SET ASIDE SPACE FOR CODE AND MESSAGE
40 DIM PROG 100
50 FOR PASS =1 TO 3 STEP 2
60 P%=PROG
70[OPT PASS
80 .START LDA MESSAGE ;GET FIRST BYTE
90        JSR OSASCI  ;ONTO SCREEN
100       LDA MESSAGE+1 ;GET SECOND BYTE
110       JSR OSASCI ;ONTO SCREEN
120       LDA MESSAGE+2 ;ETC FOR REMAINING
130       JSR OSASCI ;TWO BYTES
140     LDA MESSAGE+3
150     JSR OSASCI
160      RTS            ;BACK TO BASIC
170]
180 REM DEFINE START OF "FRED" MESSAGE
190 MESSAGE=P%
200 REM INSERT "FRED"
210 $MESSAGE="FRED"
220 NEXT PASS
230 CALL START
240 END


1060 AD 79 10 .START LDA MESSAGE ;GET FIRST BYTE
1063 20 E3 FF JSR OSASCI  ;ONTO SCREEN
1066 AD 7A 10 LDA MESSAGE+1 ;GET SECOND BYTE
1069 20 E3 FF JSR OSASCI ;ONTO SCREEN
106C AD 7B 10 LDA MESSAGE+2 ;ETC FOR REMAINING
106F 20 E3 FF JSR OSASCI ;TWO BYTES
1072 AD 7C 10 LDA MESSAGE+3
1075 20 E3 FF JSR OSASCI
1078 60       RTS           ;BACK TO BASIC
```



Program 4.
Machine code and message in reserved space on top of basic program.

indexed addressing can be used to output a message to the screen. The message is placed into consecutive locations in memory using the indirection operator described earlier. The start of the message has been given the label MESG. This is used by the LDA instruction as the base address, which, together with the continually incremented contents of the X register, enable the sequence of bytes comprising the message to be successively loaded into the accumulator and sent via OSASCI to the screen. The action within the loop of incrementing the X register contents means that the next byte in the message will be loaded into the accumulator. The process terminates when the last byte (&0D = carriage return) is encountered.

There are many uses of indexed addressing amd most microprocessors incorporate index registers, making it possible to 'point' to successive locations in memory. Moving blocks of data from one area to another or sorting lists of data are both tasks that are easily

```
100REM ILLUSTRATION OF INDEXED ADDRESSING
110OSASCI=&FFE3
120DIM PROG 100 :REM ALLOCATE PROGRAM SPACE
130FOR PASS=0 TO 3 STEP 3
140P%=PROG
150[OPT PASS
160          LDX #0
170.AGAIN    LDA MESG,X \GET CHARACTER FROM MESSAGE
180          JSR OSASCI \SEND CHARACTER TO SCREEN
190          INX        \POINT TO NEXT CHARACTER
200          CMP #&0D   \LOOK FOR END MARKER
210          BNE AGAIN  \FINISHED?
220          RTS        \BACK TO BASIC
230]
240REM DEFINE MESSAGE AREA
250MESG=P%
260$MESG="Message printed using indexed addressing"
270NEXT PASS
280END

>RUN
1023              OPT PASS
1023 A2 00        LDX #0
1025 BD 31 10  .AGAIN   LDA MESG,X  GET CHARACTER FROM MESSAGE
1028 20 E3 FF  JSR OSASCI \SEND CHARACTER TO SCREEN
102B E8           INX        \POINT TO NEXT CHARACTER
102C C9 0D        CMP #&0D   \LOOK FOR END MARKER
102E D0 F5        BNE AGAIN  \FINISHED?
1030 60           RTS        \BACK TO BASIC

>CALL PROG
Message printed using indexed addressing
```

Program 5. Using indexed addressing to send a message to the screen



```
100REM USE OF INDEXED ADDRESSING TO WRITE TO INPUT BUFFER
110OSRDCHR=&FFE0
120OSASCI=&FFE3
130DIM PROG 100
140FOR PASS=0 TO 3 STEP 3
150P%=PROG
160[OPT PASS
170          LDY#&FF
180.AGAIN    INY          \POINT TO NEXT BUFFER SPACE
190          JSR OSRDCHR \GET CHARACTER
200          JSR OSASCI  \ECHO TO SCREEN
210          STA BUFF,Y  \PLACE IN BUFFER
220          CMP #&24    \CHECK FOR $ TERMINATOR
230          BNE AGAIN   \GET NEXT CHARACTER
240          RTS         \BACK TO BASIC
250]
260REM DEFINE BUFFER AREA AT END OF PROGRAM
270BUFF=P%
280NEXT PASS
290END
 RUN
1041                OPT PASS
1041 A0 FF          LDY #&FF
1043 C8             .AGAIN   INY         \POINT TO NEXT BUFFER SPACE
1044 20 E0 FF JSR OSRDCHR \GET CHARACTER
1047 20 E3 FF JSR OSASCI  \ECHO TO SCREEN
104A 99 52 10 STA BUFF,Y  \PLACE IN BUFFER
104D C9 24    CMP #&24    \CHECK FOR $ TERMINATOR
104F D0 F2    BNE AGAIN   \GET NEXT CHARACTER
1051 60       RTS         \BACK TO BASIC
```

Program 6. Indexed addressing writes to an input buffer

```
 CALL PROG
THIS IS THE MESSAGE
$
 FOR I=BUFF TO BUFF+26:P.~I;"
";CHR$(?I):NEXT I
        1052 T
        1053 H
        1054 I
        1055 S
        1056
        1057 I
        1058 S
        1059
        105A T
        105B H
        105C E
        105D
        105E M
        105F E
        1060 S
        1061 S
        1062 A
        1063 G
        1064 E
        1065
        1066 $
        1067
        1068
        1069
        106A
        106B
        106C
```

Sample output from program 6

implemented using indexed addressing.

A further example of indexed addressing is given by program 6, which saves a series of characters input at the keyboard in memory (an input buffer). The series is terminated by entering a '$'. The buffer has been placed after the program and will be corrupted if more Basic statements are added. In practice such a buffer would be somewhat safer. The BBC machine has buffers for its own input located in page 0.

The program can be run by typing CALL PROG and a series of characters then entered from the keyboard. The following Basic statements will print out the memory address and contents of the buffer.

```
> FOR I = BUFF TO BUFF+26:P~I;
"";CHR$(?I):NEXT I
```

With indexed addressing the range of memory addressable from a base address is the next 256 bytes only. In this program there is no test of how many characters have been entered and the effect of entering more than 256 characters will be to start filling up the buffer a second time, overwriting characters placed in earlier.

Table 1 – Instructions that use Indexed Addressing

Instruction		ABS,X	ZERO PAGE, X	ABS, Y	ZERO PAGE Y
ADC	ADd with Carry	x	x	x	
AND	AND with accumulator,	x	x	x	
ASL	Arithmetic Shift Left	x	x		
CMP	Compare Memory with Accum.	x	x	x	
DEC	DECrement memory	x	x		
EOR	Exclusive OR memory with accum.	x	x	x	
INC	INCrement memory	x	x		
LDA	LoaD Accumulator	x	x	x	
LDX	LoaD X register		x		x
LDY	LoaD Y register	x	x		
LSR	Logical Shift Right	x	x		
ORA	OR memory with Accumulator	x	x	x	
ROL	ROtate Left	x	x		
ROR	ROtate Right	x	x		
SBC	SuBtract with Carry	x	x	x	
STA	STore Accumulator	x	x	x	
STX	STore X register				x
STY	STore Y register		x		

NEW from our BBC micros accessories range

GRAPHICS TABLET

Transfer your technical drawings or designs from pad to screen in seconds and in colour. 22" × 27" approx. Measures distances and areas, and moves drawings on screen ...
ONLY £75 plus £11.25 VAT plus £4.50 p&p.

BBC HARDWARE

Model 'A' 16K RAM 32K ROM. Full colour, high resolution graphics ...... **£260 + £39 VAT** (£7 p&p)
Model 'B' 32K RAM 32K ROM. 16 Colour Graphics ... **£346.95 + £52.05 VAT** (£7 p&p)

BBC Disk Interface ...... **£95 + £14.25 VAT**
BBC 'B' Upgrade Kit .... **£113.04 + £16.96 VAT** (£1.50 p&p)
BBC Single Disk Drive 100K bytes **£230.43 + £34.57 VAT** (£2.50 p&p)

14" RGB Colour Monitor (as used in the BBC Computer Programme) .... **£270 + £40.50 VAT** (£9.50 Courier)
12" Green Monitor **£82.60 + £12.40 VAT** (£6.50 Courier)
BBC Joystick ...... **£11.30 + £1.70 VAT** (£1 p&p)
Acorn GP 100 Printer ...... **£200 + £30 VAT** (£4.50 p&p)
Epson MX80 FT111 Printer, new model, dot matrix, high res. graphics, 80 or 132 char's per line ...... **£325 + £48.75 VAT** (£4.50 p&p)

Printer Cable (parallel) ...... **£13.04 + £1.96 VAT** (£1 p&p)
Tapes (Computer standard) ..... **£0.86 + £0.14 VAT** (£1 p&p any quantity)

Disks ('BBC' standard/pack of ten) ... **£19.50 + £2.93 VAT** (£1 p&p)

Books ...... Prices on application
Cover for BBC ...... **£4.99 + £0.75 VAT** (£1 p&p)
Cassette Recorders ...... **£25.22 + £3.78 VAT** (£3 p&p)
Teletext ...... available Early '83
Econet ...... available Early '83
Dual Disk Drives 800K ...... available Jan '83
Flochart Stencil ...... **£3.50 + £0.52 VAT** (50p p&p)
Acorn soft ...... List Prices
BBC Soft ...... List Prices

128K Ram Pack

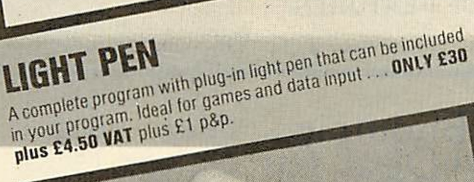
Simple plug-in memory expansion board, addressable for your own programs .... **Price £299 plus £44.85 VAT** plus £1 p&p.

FAN

For the professional man to keep the computer COOL ....
Price £33.25 plus £4.99 VAT plus £1.50 p&p

A to D

A highly accurate and PRECISE convertor for your external devices .... **ONLY £31.50 plus £4.73 VAT** plus £1.50 p&p



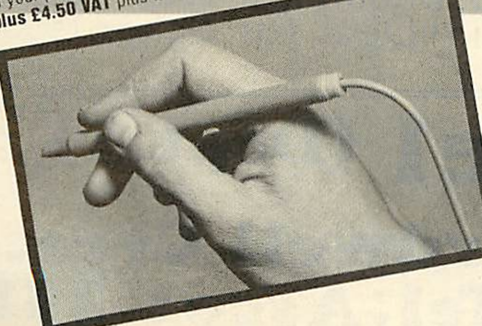
LIGHT PEN

A complete program with plug-in light pen that can be included in your program. Ideal for games and data input ... **ONLY £30 plus £4.50 VAT** plus £1 p&p.





Mail or telephone order. Barclaycard or Access Credit Card accepted.



MICRO MANAGEMENT

Micro Management, 32 Princes Street, IPSWICH. Telephone 59181

FINANCIAL

BBC Model B

GAMES

THE WORLD TRAVEL GAME





1 or 2 Players,
Choice of Game,
'Exciting, Tense, Competitive and even Educational'.
XXXXX

Travel the World; Journey by air, rail and road.
Exchange currencies; Buy souvenirs; Book tickets.
Cope with hijacks, strikes, robbery and other problems inherent with travel.

Visit countries as diverse as Russia & the Falklands.

Your aim is to collect 6 souvenirs & return to LONDON intact!

— £6.95 —

GREAT BRITAIN LTD

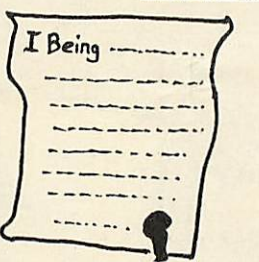


You are P.M. and Chancellor of 'Great Britain'

You must select the Party you wish to represent and your aim is to stay in office for as long as possible. You must control inflation and unemployment, maintain th exchange rate, introduce social reforms and stay popular. The game is split into sectors: country profile, shopping basket, budget day, reform opportunities, manifesto, and most important election nights (a telling time).
A COMPLEX GAME THAT YOU WILL NOT TIRE OF IN A HURRY

— £5.95 —

INHERITANCE



Gt. Uncle Arbuthnot is dead.
You stand to inherit!!

A 2 part game. Prove your financial accumen in Part 1 by investing wisely at the stock and metal markets; if desperate try the casino or the horse races. If you are successful you will enter the world of big business in Part 2. Find the secret formula for paradise cola; manufacture and market the drink; cope with strikes, fires, frauds, cash shortages, etc. Your ultimate aim is to become a millionaire! **A MAMMOTH GAME PACKED FULL OF FEATURES**

— £5.95 —

See Reviews in:
Acorn User Dec '82
Personal Computer Jan '83

Trade Enquiries Welcome
Special Deal for Schools.

Simon W. Hessel (Dept A)

15 Lytham Court, Cardwell Crescent
Sunninghill, Berks.—Tel: Ascot 25179

24hr Despatch — One Year Guarantee — Money-back if not satisfied.



A more dynamic approach

TAB is underused, ESCAPE is too close for comfort, so Joe Telford has been examining these keys. He also has some ideas on procedures

We begin by looking at some of those *FX commands which appear to be undocumented by Acorn. This month we have three to examine, which all appear to work with the 1.00 version of the operating system, and are hence upwardly compatible, ie they will work with systems numbered above 1.00, but not with 0.10.

The TAB key is probably the most underused key on the Beeb, because it appears only to copy the function of the SPACE key. In actual fact it performs the function of CTRL_I or VDU9. This is the control code for stepping one space to the right, without destroying any characters in its path.

The TAB key could be used more often if the character PRINTed by it could be altered. In actual fact the TAB key can have any character code allocated to it. Simply type *FX219,42.

Now you have allocated character code 42 to the TAB key. Whenever you press TAB, a '*' will appear on the screen. There is little value in using a spare key to duplicate the operation of an easily accessible character, already on the keyboard, but table 1 lists the VDU commands which can be allocated to the TAB key:

Second, TAB can be used for user defined characters. Type this program into the computer and then RUN it:

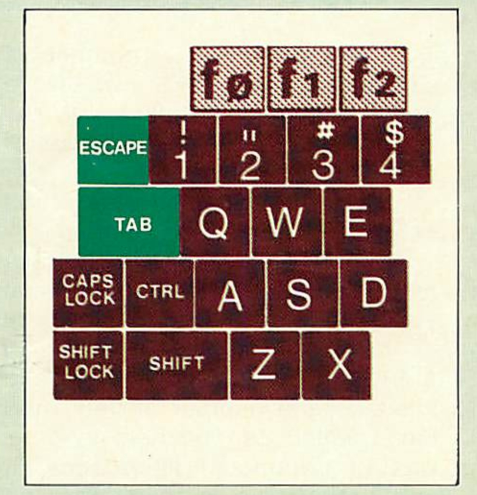
```
5 MODE 4
10 VDU23,224,3,4,4,8,8,144,80,32
20 *FX219,224
30 PRINT"Please press the TAB key"
40 PRINT"to add a new character."
```

Remember that user defined characters will not be displayed in mode 7. Hence we switch to mode 4. Line 10 creates a square root symbol and line 20 allocates it to the TAB key. Lines 30 and 40 are simply instructions at the end of the program.

TAB can be used in this way to write the square root key in print statements simply by pressing TAB whenever the particular character is needed eg:

```
50 PRINT"<TAB>36=";SQR(36)
```

Where <TAB> means 'press the TAB key.'



During a program run, the ESCAPE key is a useful addition, getting the user out of loops, etc. However, once a program is fully debugged, a good deal of effort often goes into ensuring the program will not crash if the key is pressed accidentally. Unfortunately, inexperienced fingers will find that both the TAB key and the number 1 key are close enough to the ESCAPE key to provide the opportunity for unexpected escapes.

A possible fix for this is to remove the ESCAPE key from the keyboard, but as that would require a certain hardware expertise, and would probably be messy, Acorn have kindly produced a software solution.

The ESCAPE key generates ASCII 27 and the solution is to nominate another key to produce that code. Try for example *FX220,32.

On pressing ESCAPE, nothing should happen. Now try pressing the space bar. The space bar has been set up as the ESCAPE key. It is of course important to use a key for ESCAPE which can be accessed without altering normal keyboard use. I found *FX220,0 was a suitable allocation. Once entered at the keyboard the ESCAPE function is taken over by CTRL–@, which is pressed as the result of a deliberate action rather than by accident.

To restore ESCAPE, simply type *FX220,27.

If you have a model B, machine try typing *FX254,255. Then MODE2. If you have a model A, try *FX254,0. Then MODE2. Sorry BREAK won't reset this *FX command. If you assume your machine now thinks

Allocated with	Normal key sequence	Action on pressing TAB
*FX219.7	CTRL–G	Make a beep
*FX219,8	CTRL–H	Backspace cursor
*FX219.9	CTRL–I	Normal TAB function
*FX219,10	CTRL–J	Move cursor one line down
*FX209,11	CTRL–K	Move cursor one line up
*FX219,12	CTRL–L	Clear text screen (CLS)
*FX219,16	CTRL–P	Clear graphics screen (CLG)
*FX219,21	CTRL–U	Disable VDU drivers/ Delete line in INPUT mode
*FX219,26	CTRL–Z	Restore default windows
*FX219,30	CTRL–	Home cursor

Table 1. VDU commands



```
  20MODE4
  30REPEAT
  40CLS
  50PRINT"Procedure name
   (upto 10 letters)"
  60INPUTQ$
  70IF Q$="SQUARE" THEN PROC_SQUARE
  80IF Q$="BOAT" THEN PROC_BOAT
  90IF Q$="ROCKET" THEN PROC_ROCKET
 100REM This continues ad.inf.
 190REM until they finish
 200UNTIL Q$="END"
 210END
1000DEFPROC_SQUARE
1010MOVE200,200
1020DRAW400,200
1030DRAW400,400
1040DRAW200,400
1050DRAW200,200
1060REPEAT UNTIL GET=32
1070ENDPROC
1080DEFPROC_TRIANGLE
1090MOVE200,200
1100DRAW400,200
1110DRAW300,300:DRAW200,200
1120REPEAT UNTIL GET=32
1130ENDPROC
1140DEFPROC_OBLONG
1150MOVE200,200
1160DRAW800,200
1170DRAW800,500
1180DRAW200,500
1190DRAW200,200
1200REPEAT UNTIL GET=32
1210ENDPROC
```

Program 1. A Hypothetical example

its the other model, you won't be far wrong. Powering down for a few seconds will cancel the command.

Now on to procedures. First, let's put things into perspective: I think procedures are the greatest thing since sliced bread. There are however, a few instances, where calling a procedure (or to be fair, a subroutine too) is difficult.

Look at program 1. This is a typical routine which a user might build as part of an educational program, or as a translational program or possibly even as a command language for a data base. Notice that because this program section is hypothetical and only intended to make a point, there is no value to be gained in typing it into your micro. Readers will probably have a number of similar programs with which to draw comparisons. Suffice it to say that we are dealing with the type of program which has a large number of procedures selected as a result of entering a particular command.

A short while ago this problem cropped up. I was helping a local speech therapist with some work on the Bliss symbolic language. On a BBC micro with only a cassette interface, the number of translated terms which can be held in core, was of paramount importance. It was imperative to capitalise on memory.

Bliss is based around symbols from everyday life, and because these symbols are line drawings they need to run in screen mode 4, which uses 10k of RAM. It was felt necessary to fit about 100 procedures into memory, to provide the Bliss vocabulary. The next stage was to access these Bliss terms by calling each term (procedure) from the individual words of a sentence in English. First, the current sentence was broken down into individual words. From here, a reference back to program 1 suggests each procedure calling line would look like:

```
120 IF W$="doctor" THEN PROC_doctor
```

and would take up approx 30 bytes. The lines to call the 100 procedures would therefore take up about 3000 bytes. From this it was

```
  20MODE4
  30REPEAT
  40CLS
  50PRINT"Procedure name
   (upto 10 letters)"
  60INPUTQ$
  70PROCQ$
  80UNTIL Q$="END"
  90END
1000DEFPROC_SQUARE
1010MOVE200,200
1020DRAW400,200
1030DRAW400,400
1040DRAW200,400
1050DRAW200,200
1060REPEAT UNTIL GET=32
1070ENDPROC
1080DEFPROC_TRIANGLE
1090MOVE200,200
1100DRAW400,200
1110DRAW300,300:DRAW200,200
1120REPEAT UNTIL GET=32
1130ENDPROC
1140DEFPROC_OBLONG
1150MOVE200,200
1160DRAW800,200
1170DRAW800,500
1180DRAW200,500
1190DRAW200,200
1200REPEAT UNTIL GET=32
1210ENDPROC
```

Program 2. Good idea, but it doesn't work



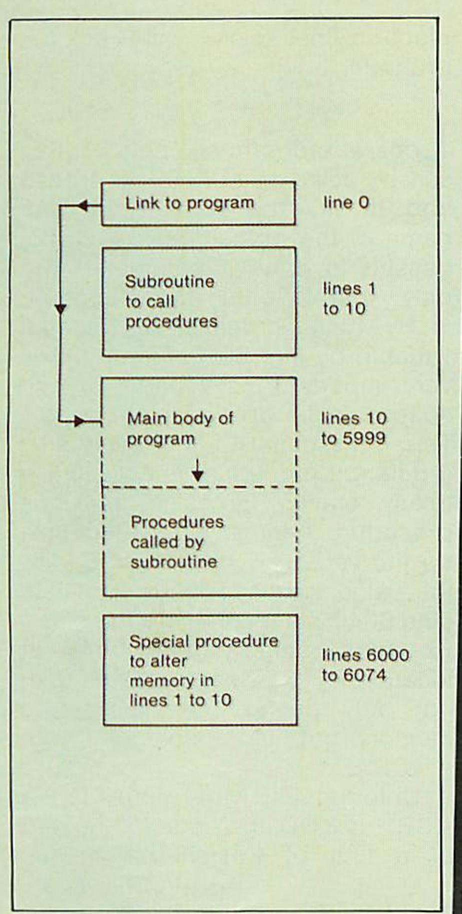


Figure 1. Theory behind program 3

a. Main program

```
20ON ERROR GOTO 200
25MODE4
30REPEAT
40CLS
50PRINT"Procedure name
  (upto 10 letters)"
60INPUT Q$
70IF LEN(Q$)>10 THEN 50
80PROC_PROC(Q$)
90UNTIL Q$="END"
100END
200IF ERR<>29 THEN 230
210PRINT "Sorry, it doesn't exist."
215REPEAT UNTIL GET=32
220GOTO30
230MODE7:REPORT:PRINT" AT "ERL:END
1000DEFPROC_SQUARE
1010MOVE200,200
1020DRAW400,200
1030DRAW400,400
1040DRAW200,400
1050DRAW200,200
1060REPEAT UNTIL GET=32
1070ENDPROC
1080DEFPROC_TRIANGLE
1090MOVE200,200
1100DRAW400,200
1110DRAW300,300:DRAW200,200
1120REPEAT UNTIL GET=32
1130ENDPROC
1140DEFPROC_OBLONG
1150MOVE200,200
1160DRAW800,200
1170DRAW800,500
1180DRAW200,500
1190DRAW200,200
1200REPEAT UNTIL GET=32
1210ENDPROC
```

Program 3. Alters memory during program run, so handle with care

b. Notice, no spaces

```
0GOTO10
1PROC_12345678901:REM
2RETURN
10REM Leave 0,1,2 alone
```

```
6000DEFPROC_PROC(F$)
6010LOCAL P
6020IF LEN(F$)<10
    F$=F$+STRING$
    ((10-LEN(F$))," ")
6030P=PAGE+15
6040$P=F$
6050P?10=32
6060GOSUB 1
6070ENDPROC
```

c. Erases previous procedure names

obvious that the program would spend its life in a lookup loop, ie it would be slow; and that 3000 bytes of procedure calling could be better used for the actual procedures themselves.

My first solution was similar to program 2. Here we simply input the procedure name and do a PROCQ$. The idea was that if Q$="SQUARE" then PROCQ$ would call the procedure defined as SQUARE.

Unfortunately this didn't work because the BBC micro doesn't support dynamic procedure calls. To call the procedure SQUARE the BBC micro expects to find the command PROCSQUARE. If instead we set Q$ to be "SQUARE" and issue the command PROCQ$, then the computer ignores the contents of Q$ and looks for a matching DEFPROCQ$, which it never finds.

What is even worse is that should your program include a line DEFPROCQ$ then the Basic interpreter delivers the body blow: 'No such variable at line XXXX'. The only way to call procedures safely would seem to be as per program 1. (Readers will now note that program 2 doesn't work.)

However, as readers will now be aware, we (the Acorn Users) aren't often put off by the mere fact that something 'can't be done'. Let's consider the facts:We need to call procedures from a single calling line: Procedures can only be called by having the actual calling word eg SQUARE placed directly after the command PROC: Calling procedures via varaibles will not work.

This means we need to place the calling word physically after the PROC command in the calling line whatever that word may be. Hence we must consider an approach which involves altering the memory contents of a line within the program, actually during the program run.

This is the first time we have considered adjustments to memory in the Hints and Tips series and so a number of points must be made:

- Direct alterations to memory often cause **crashes** if they are not properly effected.
- At the very least, programs can list in an odd fashion, or do any of a number of **unpredictable things,** should memory be incorrectly altered.
- If the program is altered at this level, the effects of further program editing, eg adding, deleting, changing lines can be **catastrophic**.
- You must continually ask yourself: 'Does this program need to work through the **Tube?** If the answer is yes, make sure it does.

Most of the above points can be solved by debugging the memory altering routines first then making sure the memory to be altered is at the beginning of the program. This means you will need a line (usually line 0) which jumps over that part of memory being altered to the main program so that any editing you do will be above the critical area of memory which is being altered during the program. Figure 1 shows a typical map for such a program in Basic.

Sections 3a,3b,3c make up a



program illustrated by figure 1. First examine section 3a.

It is important to enter this section **without any** spaces in lines 0,1 or 2. This is so that the part of memory which needs to be altered doesn't move from its set place in memory. Look at line 0. This is the link to the rest of the program. When we type RUN, line 0 jumps to line 10 which starts the program proper. The line which actually calls any procedure is line 1. It has been set up to call the procedure '_12345678901'. This is actually a dummy name which is never called. The ':REM' after the numbers is simply a marker for the physical end of the procedure name. What we need to do is alter the number '12345678901' to read things like square or traingle or oblong (or in my original program doctor etc). To do this it is necessary to find the memory location of the beginning of the string '12345678901'. Readers used to my column will have noticed my trademark of having an underline '_' at the beginning of any PROC or DEFPROC hence for me the procedure name begins with '12345678901'.

To find the location of the beginning of the string to alter we simply count characters from the program beginning. Line 0 uses 9 bytes, made up as follows:

Line number	: 2 bytes
GOTO	: 1 byte
10 (in integer format)	: 4 bytes
Length of line (unseen)	: 1 byte
End of line marker	: 1 byte

Line 2 as far as 1PROC_ uses 5 bytes.

Line number	: 2 bytes
PROC	: 1 byte
_ (underline)	: 1 byte
Length of line (unseen)	: 1 byte

And so the first character of '12345678901' is the 15th memory location after the start of the program. Because programs in Basic start at PAGE, the actual location we need is PAGE+15.

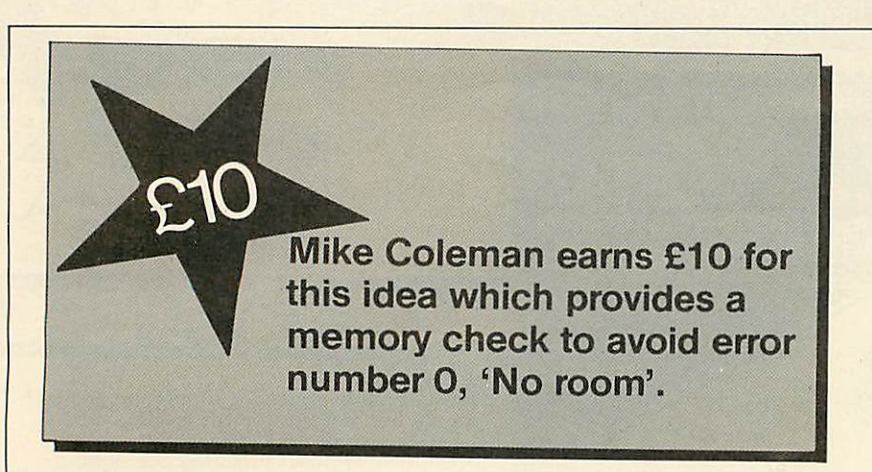
It will now be plain that we must ensure that no spaces creep in (especially at the end of line 0) to foul up our calculations. A single extra space or character out of place in lines 0 or 1 will wreck the program.

Look at program segment 3c. It is suitably defined as DEFPROC_PROC and it has the variable F$ (the name of the procedure to activate) passed to it as a parameter. The only variable within the procedure is P which is defined as a local variable by line 6010. Line 6020 is very important, as it pads F$ with spaces to fill up to 10 characters. This is essential so that every time a different procedure is activated, it totally erases traces of previous procedure names. Line 6030 sets up the variable P to be PAGE+15,' the value we have just calculated. Line 6040 places the padded string F$ into memory beginning at location P (15 bytes after PAGE). $P not only pushes the string into memory, but also concludes it with a <CR>.

Unfortunately, the code for a <CR> is &OD, the code for the end of a line of Basic. This means

page 26 ►





The ON ERROR feature of BBC Basic allows a user program to take its own evasive action if an error is detected – with one exception. Error number 0 'No room', generated by a program using up all of the available memory, is untrapable and causes the program to be aborted. Since the very nature of the error suggests that it only crops up well into a program run, it is likely to be accompanied by a grievous loss of input data. A simple solution, therefore, is to check for the condition before it arises.

Figure 1 gives a simplified picture of the RAM memory map. As a program runs, two areas are used for working storage: that above LOMEM (for dynamic variables), and that below HIMEM (for Basic stack). The appearance of of 'No room' means that, as with the Hoffnung's tale of *The Builder's Bucket*, dynamic variables going up have met Basic stack coming down! However, since the pointers to these two areas are stored in RAM locations (2,3) and (4,5) respectively, it is simple to detect this situation using the ? operator:

```
(?5-?3)*256+?4-?2
```

gives the available free store, and a function such as

```
DEFFNroom(bytes%) =
(?5-?3)*256+?4-?2>bytes%
```

produces TRUE if more than bytes% of free store is available.

Operations likely to make critical demands on store can thus be made conditional with a statement of the form:

```
If FNroom(100) THEN <proceed>
ELSE <error routine>
```

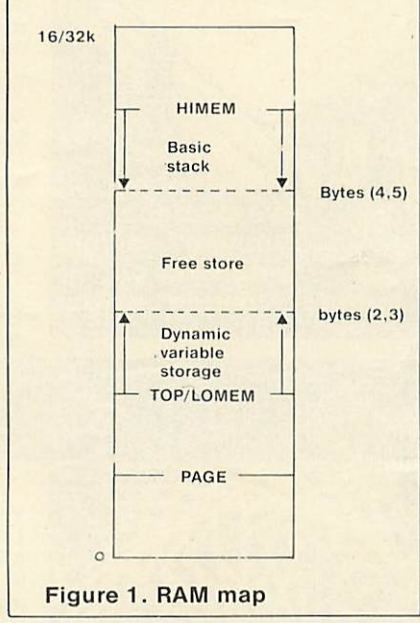


Figure 1. RAM map

The error routine need be no more than a GOTO which refers to the same destination as that of the ON ERROR statement.

□ This is a very interesting idea and it is worth noting, in passing, the use of Boolean operators in the function definition. Two points, however: first, readers should know it is also possible to check for space above the program using DIM P% -1 : SPACE = HIMEM – P% (note the space between P% and -1 is crucial); however, this takes no account of the space occupied by the Basic stack (which is typically used for procedures and functions). Second, a small point but worth observing: calling the function FNroom will take up some temporary space in the stack and so the test is very slightly conservative. IB

In last month's issue (p19), Joe Telford gave a routine to relocate programs loaded from disc. There are two disadvantages with his method, however. The first is that, since the routine *is in Basic* it is rather slow. Second, and more important, since a user-defined key has to be programmed to achieve the relocation, the method is not 'transparent' to the user. That is, any user of a program loaded from disc would have to remember (and know how) to program the key and to activate it prior to the running of any program. This is particularly unfortunate in the case of educational software. The following ideas should remove these disadvantages.

Fitting a disc interface to the BBC micro incurs a memory cost. In the standard model B, PAGE defaults to &E00: in a disc system, it defaults to &1900. This lost space is used by the disc operating system when any disc transactions occur.

Once a program is loaded from disc, and assuming it will not use the disc unit again, this lost space will no longer be required. It would be useful, therefore, to be able to shift the program down into &E00 onwards after it has loaded. This is the purpose of the program *Shift*.

Type *Shift* into your computer exactly as it's written – this is crucial. Then type *SPOOL SHIFT and LIST the program onto disc. Finish with *SPOOL.

Every program which will be

```
    0GOTO32000
32000BASE=&70:OLDL
      =&72:NUMBER=&
32010FORI%=0 TO 2
      STEP2:P%=&C00
32020[OPTI%
32030LDA &12
32040SEC
32050SBC #9
32060STA NUMBER
32070LDA &13
32080SBC &18
32090STA NUMBER+1
32100LDA #0
32110STA BASE
32120LDA #&0E
```

CUTTING THE MEMORY COST FOR THE DISC INTERFACE

required to be shifted down after loading should have *Shift* added to it. Use *EXEC to do this. Notice that your program should not use line 0 or line numbers greater than 31999.

What happens when a program, to which *Shift* has been added, is RUN? Line 0 directs the interpreter to line 32000, where the assembly program is assembled into &C00 onwards (this area is usually used for user-defined characters but it is quite safe as a temporary home for machine code). Then, a *FX call is used to put OLD <carriage return> RUN <carriage return> into the keyboard buffer (see p433, *User Guide*). *TAPE stops locations &E00 to &1900 becoming contaminated when the program is run. Next, TOP is reset to point to the end of the last line of the original program. Finally, the machine code program is called. This shifts the original program (ie the one you had before you appended *Shift* to it) down into &E00 onwards, resets PAGE and returns to Basic. At this point, the contents of the keyboard buffer will empty into the input buffer and be interpreted. OLD will reset all the pointers, and the program will RUN normally.

When you want to load a new program, press CTRL-BREAK first to reset the system.

One last point: if you want to program the break key, use:

*KEY 10 CHAIN "PROG"|M

(where PROG is the name of your program), since the reset will corrupt some of the program between &E00 and &1900.

Here are the explanatory details of the assembly program.

32030–32090 The number of bytes to be moved is given by (TOP–1)–(PAGE+9)+1.
(PAGE+9 because line 0 takes up nine bytes and will not be moved). This simplifies to TOP–9–PAGE: TOP is in &12 and &13 (low byte first) and the high byte of PAGE is in &18.
32100–32170. BASE is the first address in the memory where the program is going; OLDLOC is the first address in memory where the program (less line 0) is currently stored.
32190–32200. If fewer than 256 bytes are to be moved, proceed immediately to this move (at 32300).
32210–32290. Move X sets of 256 bytes. X contains the high byte of the number to be moved.
32300–32380. Move remaining bytes, ie the amount given by the low byte of NUMBER
32390–32410. Set PAGE to &E00.
32420–32430. Reset stack pointer.
32440. Return to Basic. (This address will need to be altered with the new Basic ROMs.)

PROGRAMMERS' FORUM FOR THE BBC MICRO

IAN BIRMBAUM sets out to improve your understanding of programming techniques on the BBC micro.

He will answer reader's questions and develop their ideas – as well as providing some of his own. But the real aim is for readers to provide the questions and answers.

At least £5 will be paid for any tip published, with £10 for those which merit a one-star award and £20 for real humdingers!

The ideas should be original, and be described clearly and fully.

Send your hints or questions to BBC Forum, Acorn User, 53 Bedford Square, London WC1B 3DZ. Please include a self addressed envelope if your contribution is to be returned. We are here to answer your problems, but cannot reply to every one individually.

```
32130STA BASE+1
32140LDA #9
32150STA OLDLOC
32160LDA &18
32170STA OLDLOC+1
32180LDY #0
32190LDX NUMBER+1
32200BEQ LOLOOP
32210.LOOP1
32220LDA (OLDLOC),Y
32230STA (BASE),Y
32240INY
32250BNE LOOP1
32260INC OLDLOC+1
32270INC BASE+1
32280DEX
32290BNE LOOP1
32300.LOLOOP
32310LDX NUMBER
32320BEQ FINISH
32330.LOOP2
32340LDA (OLDLOC),Y
32350STA (BASE),Y
32360INY
32370DEX
32380BNE LOOP2
32390.FINISH
32400LDA #&0E
32410STA &18
32420LDX #&FF
32430TXS
32440JMP &8EB4:]NEXT
32450*FX138,0,79
32460*FX138,0,76
32470*FX138,0,68
32480*FX138,0,13
32490*FX138,0,82
32500*FX138,0,85
32510*FX138,0,78
32520*FX138,0,13
32530*TAPE
32540HIGH=TOP-775:
      ?&12=HIGH MOD256:
      ?&13=HIGH DIV256:
      HIGH?-1=&FF
32550CALL&C00
```

This program regains the memory reserved for the disc filing system



LIMITS TO MULTIPLE PROGRAMS

£5 MICHAEL MURRAY WINS £5 FOR THIS TIP

This tip relates to the article 'Multiple program storage' published in the November issue of *Acorn User*.

Memory limitation is most likely to be encountered if one of the programs uses high resolution graphics. If programs have been entered satisfactorily in a mode where HIMEM has a high value, but one of the programs tries to use a mode for which the value of HIMEM is lower than the TOP of the final program, the run will stop and the computer will display the error message.

```
Bad mode at line (line number)
```

If this situation doesn't occur it is still possible to run out of memory if insufficient space has been left to store all the variables. These are stored between TOP of the final program and HIMEM, which will vary from program to program depending on which mode each uses. If a program attempts to dimension an array for which there is too little room it will stop and display:

```
DIM space at line (line number)
```

If the program creates too many variables while running and runs out of room it will stop and display:

```
No room
```

If any of these memory limitations are encountered there are three options available. The first is not to run the program which causes the problem. Second, change the mode of the offending program to free more memory. Finally, delete the upper programs one at a time until all those remaining will run.

To delete the last program go to the next to last and enter OLD. This does not actually delete the last program but allows it to be overwritten by variables created by the lower programs in the next run. If this is done it may be necessary to reprogram the function keys or otherwise alter the value of LOMEM which is passed to each program.

☐ **It should be pointed out that pressing ESCAPE will reset TOP to the top of the current program (indeed any 'error' will do this). One way round this in version 1.0 OS is to multiply the escape key with *FX229,1, but this may not always be convenient.**

PRINTER LINE-FEEDS: NO HASSLE

Let's end this month with a very short tip. Someone sent me a complicated technique for dealing with a printer which has no auto line-feeds. There is no need for this complication: one *FX call will do the trick! Execute *FX6,0. This defines control code 0 as the code which is suppressed when output to the printer.

Normally it is code 10 (line-feed) which is suppressed; since code 0 does nothing anyway, the effect of suppressing it has no effect on the printer. This means all line-feeds will be sent to the printer whenever one is sent to the screen. And since a line-feed is sent to the screen each time it executes a carriage return, this will compensate for the lack of an auto line-feed on the printer.

HINTS & TIPS

A more dynamic approach continued from page 22

problems can occur particularly on listing the program. The fix for this is to replace the <CR> character with a space. This is done by line 6050 which says: 'Go 10 bytes past location P and place a space (character 32)'. Referring to line 1 of the program:

```
1PROC_12345678901:REM
```

it can be seen that the 10 byte string $P covers the numbers '1234567890'. The last '1' is first replaced by the <CR> code, then by the space code. After passing the string 'SQUARE' to PROC_PROC it is transferred to line 1 by directly amending the memory locations of line 1, so we would see:

```
1PROC_SQUARE          :REM
```

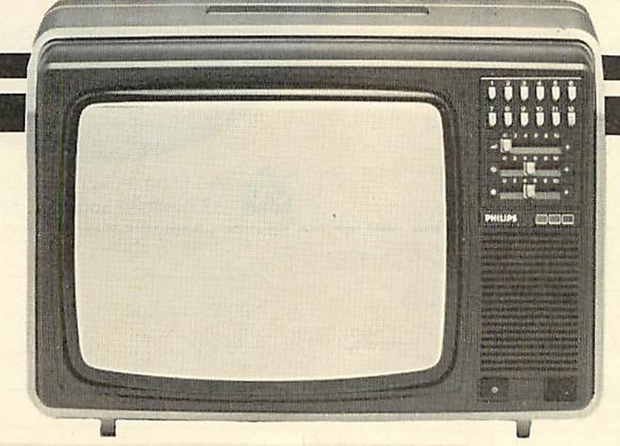
if we listed line 1. Notice the padding of spaces up to the REM. Finally, line 6060 forces a GOSUB 1, and the micro executes the now amended code at line 1. The result is that a procedure is called and after this procedure is ended, the RETURN of line 2 sends us back to line 6070, the final ENDPROC of the procedure calling route. It returns execution to the main body of the program, section 3b. (Section 3b consists of lines 20 to 230 which actually get the procedure names from the keyboard (lines 50, 60) and then activates the driving procedure (via line 80). Lines 200 to 230 take care of errors caused by incorrect entry of names, or names which do not exist. Any procedure after 1000 can be called simply by typing the procedure name. Hence it is possible to have a large number of test procedures easily accessed.

By entering program sections 3a,3b,3c at the keyboard this program will run.

Make sure it is correctly entered. Typing a procedure name at the 'Procedure name' prompt will draw the shape required, or in the case of procedures which don't exist, the message 'It doesn't exist' will be displayed. Readers may wish to expand this simple program along the lines of different shapes. If so then it is only the bare bones of an idea which needs much more thinking out. In any case, readers experimenting with the program will no doubt develop particular applications.

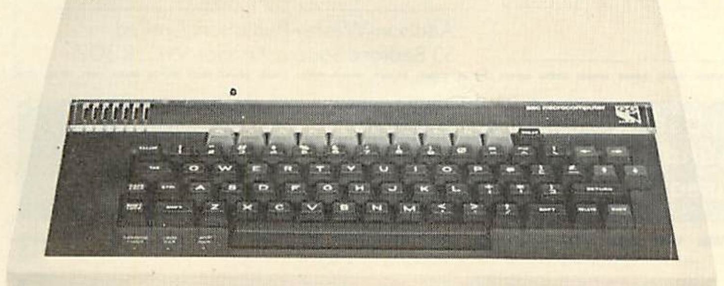
● *Next month we look at some do it yourself hardware, with software support. . . Writing the light fantastic.*

Cumana drives BBC Micro best!



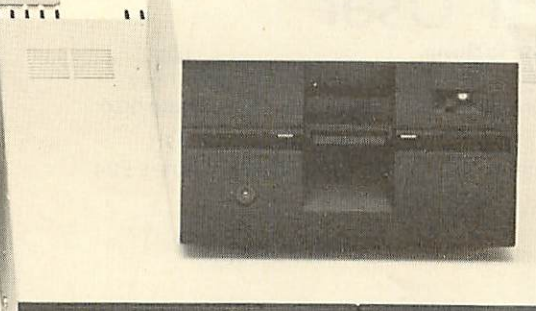
CUMANA DRIVES + OWN POWER SUPPLY = BIG PLUS FOR BBC MICRO USERS . . . JUST FOR STARTERS

. . . PLUS NO HASSLE 12 MONTH WARRANTY



The 'ice on the cake' is that, because the Cumana Drive has its own power supply, it can be used with many other Micros when connected via the appropriate cable. Cumana supply a Drive connecting cable which has a standard 34 way edge connector plus 34 way BBC connector in the same cable length. This allows the Cumana Drive to be connected to numerous makes of micro without the need to change connecting cable. And the Japanese manufactured disk drives are quiet and utterly dependable.

CS50A	Single sided 40 track TEAC drive in a cabinet with own power supply 100K	£199
CD50A	2 single sided 40 track TEAC drives in a cabinet with own power supply 200K	£369
CS50E	Single sided 80 Track TEAC Drive with cabinet and own power supply 200K	£265
CD50E	2 single sided 80 Track TEAC Drives in cabinet with own power supply 400K	£495
CS50F	Double sided 80 Track TEAC Drive with cabinet and own power supply 400K	£345
CD50F	2 double sided 80 Track TEAC Drives with cabinet and own power supply 800K	£619
	2 drive Cable for BBC Micro	£15
	2 drive Cable for BBC Micro plus TRS80, Video, Genie etc.	£18
UPGRADE KITS	A → B	£90





Please add VAT to all prices

DEALER & EDUCATIONAL ENQUIRIES WELCOME – GENEROUS DISCOUNTS AVAILABLE

CUMANA LTD

Unit 1, The Pines Trading Estate, Broad Street, GUILDFORD, Surrey. GU3 3BH. Tel: (0483) 503121. Telex: 859380 CUMANA.

To ensure regular and early delivery of Acorn User, send this form (or a copy) to: Acorn User, MAGSUB (Subscription Services) Ltd, Ground Floor Post Room, Oakfield House, Perrymount Road, HAYWARDS HEATH, West Sussex, RH16 3DH.

Acorn User

Direct Subscriptions

Please open one year's subscription to *Acorn User*.
Annual subscription rates (please tick appropriate box):

☐ UK £15 ☐ Europe £18 ☐ Middle East £20
☐ The Americas & Africa £22 ☐ All other countries £24

Name..

Position..

School/College/Company..

Department..

Address..

....

.... Post Code ....

AU7

Preferred Method of Payment
Please complete the appropriate section and delete where necessary (*).

UK Subscribers
I enclose my cheque/postal order* for £..............
payable to Addison-Wesley Publishers Limited.

Overseas Subscribers
I enclose my cheque/international money order/sterling bank draft* for £..................................
payable to Addison-Wesley Publishers Limited.

Credit Card Payment
Please debit my Access/American Express/Barclaycard/Diners Club/MasterCard/Visa*.

Account No.
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]

Signed.................... Date....................

Important Note
If you are paying by credit card, the address you give for delivery of *Acorn User* must be the same as the address to which your credit card account is sent.

Send this form, with your remittance, to the address above.



8 page pull-out

GET ORGANISED

In this regular series for primary schools Paul McGee, Inspector for Schools in Croydon, with a variety of contributors, advises teachers on how to make the most of the micro in the classroom.

This month Charles Bake, a teacher who has taken charge of the BBC micro at Woodside School in London gives his ideas on how the computer should be used and organised. It may be harder to get teachers motivated than pupils.

The spring term has just begun and some primary schools will have received their micro – and along with it the MEP Microprimer training package. But even before these arrive, thought must be given to how the micro will be used.

A system will be needed to govern how the micro will be used in the school as a whole, in each classroom, and with each pupil. Should there be a computer room? Are large working groups better than small ones? How do teachers know which software to use, and which children have already used it? These are just a few of the questions which spring to mind.

Then there is the problem of persuading all teachers to use the computer – and how to overcome the fact that some children may already know more about computers than their teachers.

No article can answer every question, but Charles Bake has already met most of the problems. Here, he provides his answers and advice.

Photos: Malcolm Aird





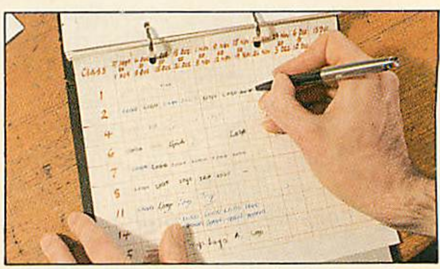
PRACTICAL DECISIONS AND IDEAS FOR THE CLASSROOM

Where to put the micro in your classroom may well be decided by the availability of sockets. However, there are two important factors to consider. First, all the pupils in a class must be able to see a screen when you are demonstrating a new program. At least one large television is needed as well as a monitor to enable those at the back of the class to read the screen.

Second, when the micro is being used by small groups, the television should be turned to face away from the class so it does not distract the other children. The monitor can be placed so that only the teacher can see it, to supervise the children.

I have not found the micro to be a distraction in class, as the children accept it as another piece of educational hardware and do not give it another thought . . . until it's their turn.

One of the benefits of using a microcomputer is that, like science experiments, it can generate interaction and constructive discussion between pupils. This works best if the children work in groups of two or three rather than individually. Ability groups may prove better than simply friendship groups, otherwise one child may do all the thinking for the less able ones, who then gain little benefit. At the start of the term, my second-year children are paired with first-years to give the younger ones confidence.


Keeping a record of who's doing what

One of my important tasks as co-ordinator of computer activities is to keep a record of work done by each class at the computer. I have a note of the programs being used by each class on a grid sheet in a ring file. This is not, as one member of staff suggested, because I wish to act as 'Big Brother', but so I can suggest which programs would be suitable to complement the one they currently use. Also it is handy to know which classes have used which programs. At the class level, teachers keep a variety of more detailed records to remind them of the stages reached by each pupil and clearly some form of standardisation would be helpful.

Children in my class use the back of their mathematics exercise books to record their work. As can be seen from the example, children are given a problem, note the commands they use in attempting to solve it and then draw the results (page 32).

Admittedly this is a rough and ready form of record-keeping: the children may, and do, forget to note the commands they use or may record them inaccurately. A printer would be a great advantage here as a hard-copy of their work could be used for discussion with the children concerned.

MICROS ON THE MOVE

AT Woodside there is no easy access to all the classrooms, and the building has long, narrow corridors with stone steps. Despite these problems we do not keep the micro in one room but trundle it about the school on a trolley. This doesn't always travel in the direction it's pointing – rather like a supermarket basket it seems to have a mind of its own – but nevertheless it has provided the necessary mobility. It also has the added advantage of having a bank of sockets attached to its side which makes the job of plugging everything in far easier than if leads and extension cables were trailed all over the room. A ramp, constructed by a helpful parent, is used to negotiate the steps.

The trolleys we used are from Selmor Engineering which are cheaper than the ones displayed at MEP Regional Information Centres. Your LEA may supply the trolleys as an item of furniture, or because it makes it easier to store the equipment in a secure place. We take the precaution of locking the micro away each night in a secure storeroom.

I am responsible for the micro and so every morning before school, I set it up. At lunch time, I ensure it is ready for the afternoon. Sometimes, especially after a busy dinner hour, I simply haven't the time to do the job and I have now trained several groups of children to do this job for me. They can connect it up properly whilst observing all the necessary safety precautions. This helps accustom the children to the computer and makes them more confident when handling it.

ORGANISATION AND TIMETABLES

When schools acquire micros they have to decide how to use them throughout the school and in individual classrooms. Decisions have to be made about whether the micro comes to the children or the children go to the micro.

Why not a computer room? If my school had opted for this, the micro would not have been so easily accepted, or as frequently and readily used. Teachers would probably have been loathe to traipse their classes along to pay homage, especially if each child had to take some work along to do whilst the privileged few were allowed to touch the machine.

But even before the micro arrives, it will be necessary for staff to discuss the sort of impact it will have on the curriculum and the philosophy of the school. For example, agreement will have to be reached as to whether it is desirable that all children in the school are given the opportunity to use the micro, or whether its use be limited to, say, the third and fourth year pupils. At this stage it will be necessary to draw up a timetable for the whereabouts of the computer, which can always be changed after a few weeks' experience.

In my school there is one micro and each class has use of it for half a day per week. This system was inaugurated by the head who thought every child should be exposed to the micro. It works well, but often teachers feel there is insufficient time to let children explore programs and with large classes it can be a case of 'right your five minutes on the micro starts . . . NOW!'

When our second micro, the DOI one, arrives we plan to have the two machines concentrated in one year group, or in one half of the school, for long periods – say half a term at a time. In that way the computers' timetable will be more flexible and teachers will have a better idea of how to use the micro during the week. It may be, for instance, that one class has the computer for two or three days at a time to fit in with projects or reinforce the current scheme of work. Or perhaps one group of chldren could be allowed to spend most of a day following through a program in detail instead of having to make way for their classmates.



BUILDING UP A SOFTWARE LIBRARY

One of my responsibilities is to build up a software library for the school. This means creating a catalogue of the software we have, writing a brief description of each program for the catalogue and, storing the cassettes. I have also had to devise a way of assessing software to decide whether we should purchase it for the school, as well as coming to the rescue when a program crashes.

I keep the software catalogue in a ring folder at present, although we may need to use a card index when we have more software. A brief synopsis of what each program has to offer is included as well as title, source, and recommendations for suitable age groups.

The cassettes, now some 35, are stored in polythene food boxes, and only those in current demand travel with the computer.

We record our programs on C10 or C15 tapes with each program on both sides of the cassette. This may seem extravagant, but it means you do not have to rewind a tape to load a program and there are two copies in case one gets damaged.

Most of our programs were obtained from the local teachers' centre, but some I typed out myself from published listings. With the current paucity of good educational software this is something a teacher has to be prepared to do.

We have devised a form to help us assess software and I would advise any school to use something like this.

Of course bugs do occur in our programs, especially those which I have typed or modified, and sometimes a child manages to press a combination of keys which confuses the micro. For instance, one program requires pupils to input 3/4/5W, but if they press 3 4 5W they get an error message. However, some classes now possess one or two children who can act as trouble shooters and will do the job for me. The normal procedure to recover from software failures is to press ESCAPE, if the program is error trapped, or else BREAK and start again. Teachers and pupils quickly learn to do this for themselves.

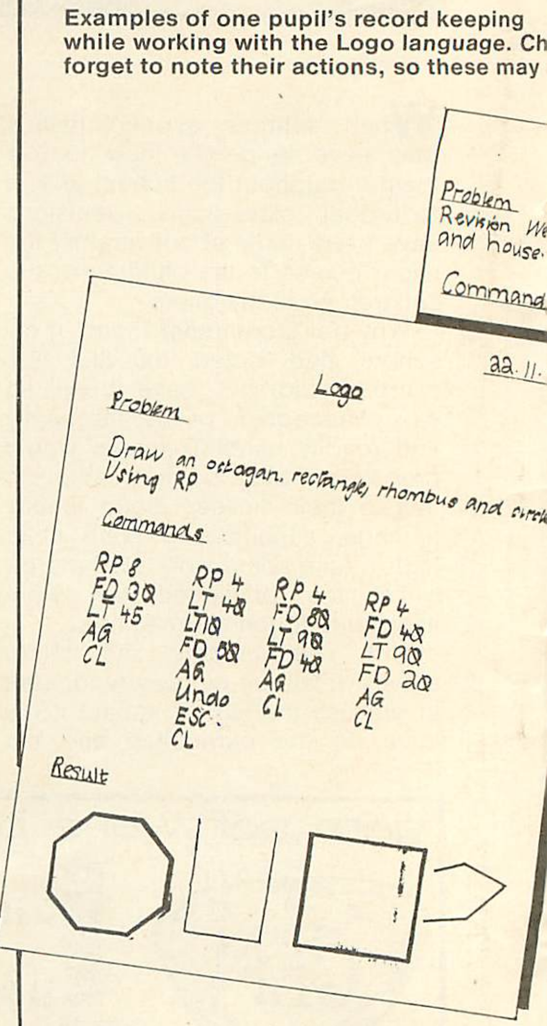
Whichever programs a teacher decides to use there must be careful thought about the preparation needed and suitable follow-up activities. For example, there is no point in launching straight into a program which requires the pupils to use a system of co-ordinates to find some imaginary beast unless pupils know the convention that numbers on the 'X-axis' precede those on the 'Y-axis'. Having played the game and caught the beast, they may then need further practice to reinforce what they have learnt. Some programs are more suited to large rather than small groups. A game such as *Animal* which requires children to generate questions to differentiate between kinds of animals seems to work best when many childen are involved. Similarly programs based on the cloze procedure which develop the pupils' ability to predict missing text, work better with groups of at least half a dozen pupils.

Programs such as *Logo* can introduce young children to concepts which previously might have been thought the province of top juniors only. For instance first and second year juniors can use 'turns' measured in degrees as part of shape drawing problems even though they would not normally cover this concept in maths lessons until they are older.

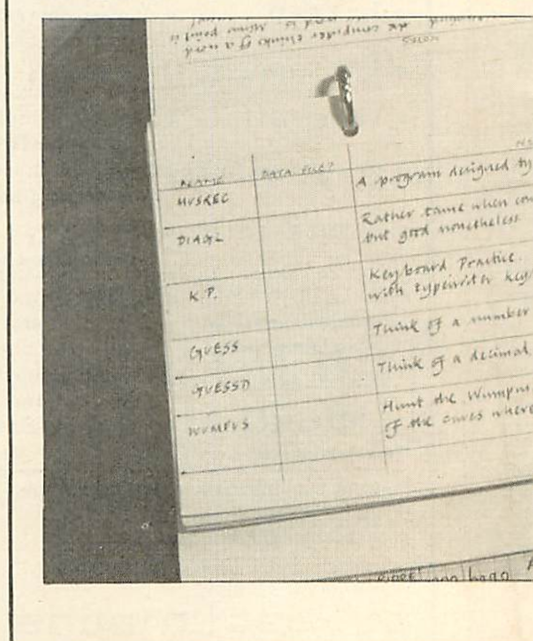
Similarly, text editing programs can be used to assist children with both factual and creative writing, often resulting in an increased quantity and quality of their output.

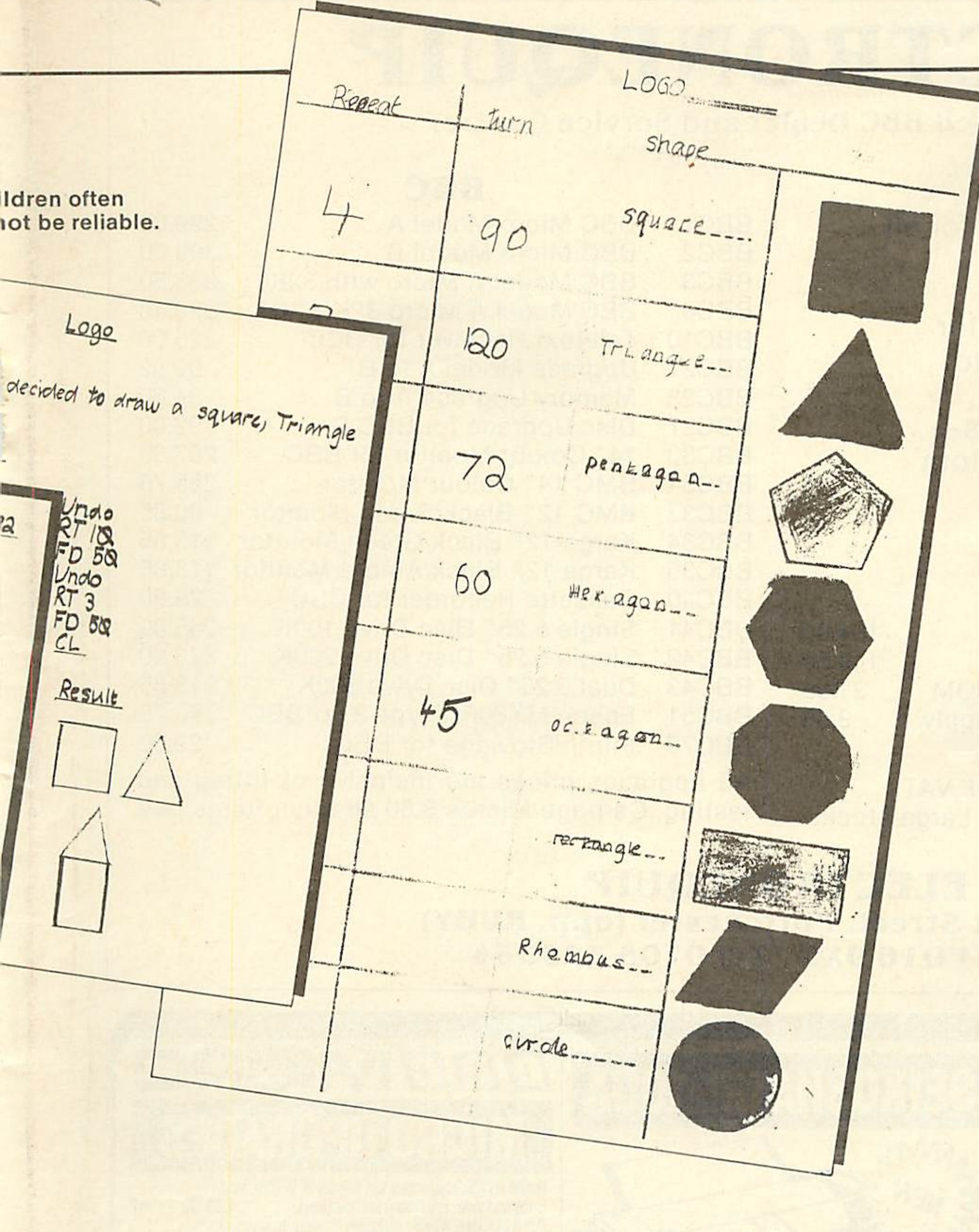
Teachers must, however, beware of bad software. Bad software can lead to even the best teachers giving bad lessons.

Examples of one pupil's record keeping while working with the Logo language. Ch forget to note their actions, so these may

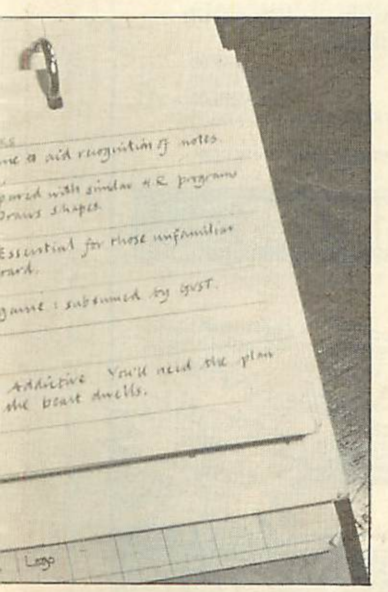


One page from the software catalogue. This just gives a short note on each progra





...ldren often
...not be reliable.



...m available.

REMEMBER TO...

- **Use large screens for class lessons.**
- **At other times, screens must face away from the class.**
- **Organise into ability groups.**
- **Prevent the addicts from monopolising the machine.**
- **Ensure the girls and boys get equal time at the micro.**
- **Keep records of work done.**

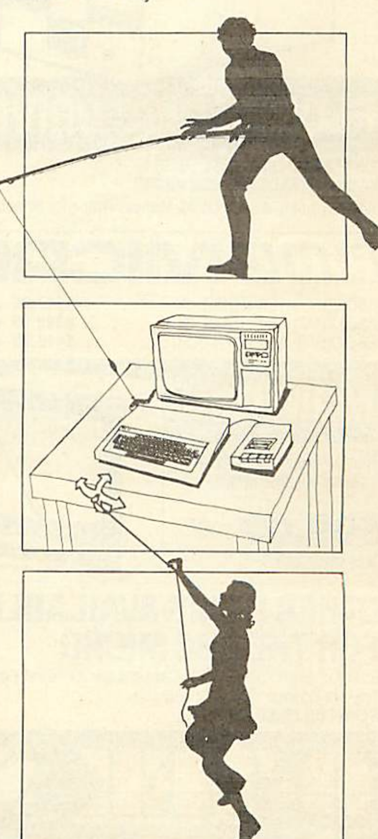
DEALING WITH THE REAL ENTHUSIASTS

IN ANY school there is likely to be an enthusiastic elite of pupils who wish to use the micro at every available opportunity. This enthusiasm can be channelled to good effect through a micro club.

At Woodside I run such a club. Interested children meet each Monday to be shown a technique in a program like *Logo* or, if they wish, can try their hand at simple programming in Basic.

During subsequent dinner hours pair of club members are permitted to use the micro to follow up the work presented on the Monday. As might be expected, I have a waiting list of children that will take me well into 1983.

It is important to make sure that these enthusiasts do not monopolise the micro in class. You may be able to channel their energy into helping you to look after the computer, trouble-shooting and assisting with the software library.



SPREADING THE WORD TO INFANTS AND PARENTS

Parents usually show a lot of interest in microcomputers. Some may want advice on what sort of micro to buy for their own use. Others may want convincing that the micro will not detract from school essentials such as learning tables. Others may feel anxious at the thought of yet another school activity with which they are unfamiliar and may themselves feel threatened by this 'knowledge gap'.

We ran an evening at Woodside to show the parents our new acquisition. This had to be organised in two sessions due to the enthusiastic response by parents. It included demonstrations of some of our programs and videos of *The Computer Programme*. Parents were encouraged to use the computer themselves and, though reluctant at first, soon had to be prised off the machine.

I would advise any school to arrange such a parents evening within a few months of starting work with their micro. The increased interest can prove very useful if you are trying to raise funds for more equipment such as printers.

Some of my colleagues who work in the infant department have reservations about using microcomputers with the very young. To see just what can be achieved, I intend to arrange for groups of infants, with an accompanying teacher, to come to the junior department during one lunchtime per week to try out programs.

This will give the teacher the opportunity to gauge the benefits of using a micro with infants and help the children themselves to enjoy a new experience.

As well as demystifying the computer for parents and the infants department, it is important to make sure that all your colleagues are at ease with the micro. Anxiety may arise if some of the children are already conversant with handling a micro (perhaps because they have one at home) and appear to threaten the teacher's usual role as the fount of all wisdom. This anxiety will not be reduced if staff return from their training courses and set themselves up as instant experts. Unless this unease is recognised and dealt with, staff may well be unwilling to have the micro in their classroom.

One way to confront the problem is to encourage staff to become familiar with the machine just as they would a new washing machine or a new car. This can be achieved by ensuring that they use it and 'see what happens if . . .'.

'Parents soon had to be prised off the machine'

They should either be encouraged to take the micro home or be given sufficient time in school to play around with it. I set up our micro at 8.15 on Monday and Tuesday mornings so staff can use it before school. I make sure I am available to sort our any problems and to instruct staff in the use of new programs.

Ten simple points to look out for

- Try to involve colleagues, but do not oversell the benefits of the micro.
- Site the microcomputer away from the sink micros do not like water.
- Acquire a trolley to make moving the micro and the television, or monitor, easier.
- Find out and obey your local education authority's safety regulations about the length of mains cables and the types of adaptors.
- Make sure girls and women teachers are encouraged to use the computer.
- Prevent the computer junkies from hogging it, and turn their enthusiasm towards something constructive, such as helping out and looking after the machine.
- Ensure that *all* children use the computer, particularly if parents provided the money and will inevitably be asked to provide more.
- Involve parents in what you are doing with the computer – but beware of computer experts who know nothing of education.
- Encourage pupils to set up the machines themselves.
- Consider your next purchases carefully – should it be discs, printers or more computers?

NEXT MONTH: MATHS AND COMPUTERS IN PRIMARY SCHOOLS



Introducing . . . Fred Jim and Sheila

Three pages which pose as ordinary locations in the Beeb's memory actually control the flow of information to the 1MHz bus and between the machine's internal control devices. Paul Beverley knows where to find them

This is the first of two articles looking at what is known as memory-mapped input/output, on the BBC micro. The first article introduces the idea and talks about the software (programming) side. In the next article, I shall put forward a few hardware ideas – explaining what is involved, in terms of electronics, in connecting up various devices to the interface known as the 1 megahertz (1 MHz) bus.

The basic operation of any computer consists of accessing and manipulating data contained in numbered memory cells. Some of these numbered cells contain fixed information which the processor uses, but is unable to change. This is called read only memory, or ROM and contains the machine code programs that determine the way the system as a whole operates. There are other memory locations which the processor can not only read, but also alter. This is called read-write memory, or more commonly random access memory (RAM). It is used to store the user's programs and any transient data which the user, or the processor itself, needs.

In the 6502 micro which the BBC machine uses, data takes the form of eight-bit bytes, so we talk of it as an eight-bit processor, but when the processor addresses the memory locations it uses 16 bits. This means that it can talk directly to 2^{16} (65,536) memory locations. (To abbreviate this, we use the fact that 65,536 = 64 x 1024, and 1024, since it is approximately 1000, is known in computer jargon as 1k.) So an eight-bit processor can have 64k of memory made up of any mixture of ROM and RAM. By a process known as paging you can put in more than 64k of memory. You have sets of memory chips and use a special chip to switch between the different sets. I will deal with this idea in the next article and explain how to attach up to 64k of RAM onto the 1MHz bus.

The model B has 32k of RAM and 32k of ROM. Half the ROM is used for the Basic interpreter and assembler, while the other 16k is used for the machine operating system (MOS), a set of machine code routines which do the housekeeping jobs, whichever high level language is being used – Basic, Lisp, Forth etc.

It is possible to have more than 32k of ROM on board since part of the ROM space is paged. Acorn have assumed that the operating system calls will be needed whatever you are doing. Therefore, it is the Basic interpreter which can be exchanged for other software (or rather firmware as it is suplied in ROM). There are three spare sockets on the main printed circuit board, into which you can put other ROMs, but if you have a disc system you will find that one of them is already filled by the disc operating system software. These ROMs can contain other language interpreters, or word processing facilities, so that you can say *PASCAL, *LISP, *WORDWISE etc, to switch from language to language. They can also contain extra commands, such as graphics routines. These extra commands can be called by names, such as *SPLINE, *FILL or *DISASSEMBLE. Finally, they can be actual applications programs. You type in *ROM and then, as normal, LOAD "GAME", or whatever the program is called. The operating system searches through all its ROM to see if it has got a program of that name, and if so, loads it into RAM.

To say the MOS is 16k is not quite true. There are three pages within it (one page = 256 bytes) where the processor is not, in fact, accessing the ROM, but is talking to actual physical devices and passing data to and from them. This is what is called memory-mapped input/output. The processor



Addresses	Device	Comments
FE00 - 01	6845 CRT controller	Can be accessed directly using VDU 23,0 (see page 385 of the *User Guide*)
FE08 - 09	6850 ACIA	Parallel to serial conversion for cassette interface and RS423 serial port
FE10	Serial ULA	
FE20 - 21	Video ULA	Colour palette
FE30	74LS161	Controls paged ROMs
FE40 - 4F	6522 internal VIA	Various internal functions
FE60 - 6F	6522 external VIA	PA port used for parallel printer. PB is the user port. Timers used for games
FE80 - 84	8271 floppy disc controller	
FEA0 - A3	68B54 advanced data link controller	Econet communications interface
FEC0 - C2	UPD 7002 analogue to digital converter	

Table 1. Addresses of devices attached to Sheila

thinks it is talking to various ordinary memory locations, when in fact the information is being passed to and from different types of electronic control devices or external devices, such as printers or plotters. The three pages have been given names – Fred, Jim and Sheila. Fred and Jim are the two pages used for the 1MHz bus, whilst Sheila is used to access all the control devices within the machine itself. This means you have to be careful when working with Sheila. Poking the wrong numbers into the wrong places can cause the whole system to crash fairly convincingly.

The actual devices which Sheila controls are listed in table 1. With the exception of the external 6522 versatile interface adapter (VIA), it is unlikely that you will want to access any of them directly. The 6845 cathode ray tube (CRT) controller and the video uncommitted logic array (ULA) are being accessed every time you PLOT, DRAW or PRINT, or when you use a VDU command. The ACIA and the serial ULA are used whenever you SAVE or LOAD, in the cassette system, though you do not access them directly.

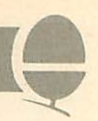
If you have a disc system you will be using the floppy-disc controller chip, but you are in fact also using the LS161 since this is used to select the paged ROM where the disc software is held. The internal 6522 VIA is used for all sorts of internal control functions, most of which you are probably not aware of. For example, it is one of the timers in this VIA which generates the interrupts for updating the pseudo-variable TIME, and also the interval timer. One of the ports of the external VIA is used for the parallel printer interface, whilst for serial printers, it is the serial ULA and the ACIA which are being used. Finally, there is the advanced data link controller, which is used for communications with the Econet system, and the analogue to digital converters which are accessed by the ADVAL function. In each of these cases, you are not addressing the devices directly, but rather using Basic commands which call routines from the operating system.

FC00 - FC0F	Test hardware
FC10 - FC13	Teletext
FC14 - FC1F	Prestel
FC80 - FC8F	Test hardware
FCC0 - FCFE	User applications
FCFF	Paging register

Table 2. Suggested memory allocation for Fred. (Jim is used for paged memory in conjunction with FCFF.)

If you want further information on any of these devices, there are data sheets available for the standard chips, such as the 6522, 6845, 68B54, 8271 and 6850. Most of these would need a whole article to themselves (in some cases a series of articles) if you wanted to describe how they worked.

Fred and Jim allow you to attach a wide range of extra peripherals to the standard machine. The first of these to be commercially available from Acorn will be the teletext receiver, followed by the Prestel receiver, possibly an EPROM programmer, and no doubt various others. Certainly, you will find a number of other manufacturers producing devices such as bit pads and so on, which will use the 1MHz bus. Table 2 shows the suggested memory usage for Fred and Jim. The test hardware referred to, includes things like the progressive establishment tester (PET) which dealers can use to help fault-find an apparently dead machine, and the final inspection



Name	Address range	OSBYTE CALL	
		Read	Write
FRED	FC00 - FCFF	&92 (146)	&93 (147)
JIM	FD00 - FDFF	&93 (148)	&95 (149)
SHEILA	FE00 - FEFF	&95 (150)	&97 (151)

Table 3. OSBYTE calls for accessing memory mapped input/output

tester (FIT) which is used to give a final check-out to an apparently healthy machine.

If for any reason you want to talk directly to any of the devices attached to Sheila, or devices attached to the 1 MHz bus, you will have to learn how to do so. Since they are memory-mapped, you could simply inspect, or alter the contents, of any of the memory locations by using the byte indirection facility, ie by saying ?&FD00 = &FF, or PRINT ?&FC01, or whatever, This is explained in the *User Guide* on page 468, but in fact it is not the proper way to do it. That is to say, if you do use byte indirection, or word indirection for that matter, your programs will no longer work when you attach a second processor to your machine. Routines have been provided within the operating system (from versions 1.0 onwards) for accessing memory-mapped input/output, and by using these routines you are assured your programs will not have to be rewritten when you attach a second processor.

There are a couple of minor disadvantages though to using these routines. First, execution times are longer than when using direct access, but this would only be noticeable if you were working entirely in machine code. Second, to use them, you have to know a little bit about assembly language on the 6502 processor, but don't panic, it's not that difficult and anyway it's good for your overall understanding and appreciation of the machine.

The routines we want to use are included in the set of routines known as the OSBYTE routines, OS standing for operating system, and BYTE referring to the fact that they can only be used to pass single bytes. Routines for transferring numbers of bytes at a time are referred to as the OSWORD routines. When you want to go from Basic into a machine code routine you can use one of two keywords, CALL and USR. CALL is used when you want to link into a routine but don't need to return any values from the routine, so it can be used for outputting bytes. However, for inputting, you have to use the USR function. You would tap something like:

```
input% = USR(&FFF4)
```

which means, 'Call the machine code routine which starts at &FFF4, and put the value obtained into the variable "input%".

The 6502 has three internal registers to which CALL or USR can pass single bytes: the accumulator, and the X and Y registers. Before you enter the routine, Basic takes the values from the least significant bytes of the integer variables A%, X% and Y%, and puts them into the appropriate registers. At the same time, the least significant bit of C% is transferred into the carry flag, which can then be tested within the routine.

To use any of the OSBYTE routines you CALL &FFF4, or use USR(&FFF4), and exactly which of the routines is called is determined by the number held in the accumulator. If A%=1, the first OSBYTE routine will be run. A%=2 would give the second routine and so on. Then further information can be passed into the routine by using the X and Y registers. The routines we want to use for memory-mapped input/output are numbers 146 to 151, and their use is summarised in table 3. The actual address which is accessed within the specified page is determined by the contents of the X register, so that if X% was made equal to &20 and you called OSBYTE 150 or 151 you would be accessing the video ULA at &FE20. The Y register is then used to contain the data being transferred either into or out of the routine.

Because so many of these OSBYTE routines are likely to be needed by the average programmer, a function is provided, within the operating system, which allows users to call those routines which involve writing rather than reading, either from the keyboard or in a program, by using the *FX command. Up to three parameters can be passed, corresponding to the A, X and Y registers in that order. If either the Y value, or both the X and Y values, are ommitted, they default to zero. For example, *FX5,2 which selects the serial printer option, uses OSBYTE call number 5 and passes a 2 to it in the X register.

To show exactly how to use these particular OSBYTE routines it is probably easiest to look at a few examples.

Suppose we want to output the number 136 to the user port, then since the address of this port is &FE60, we choose OSBYTE 151 (= &91) and make the X register equal to &60 (96) and the Y register equal to 136. We can do this by saying either:

```
A% = &91 : X% = &60 : Y% = 136:
  CALL &FFF4
```

or:

```
*FX 151,96,136
```

The advantage of using the FX call is that it is shorter, but the disadvantage is that you cannot use any variables. In other words, you could not say:

```
FOR N% = 0 TO 255 : *FX 151,96,N% :
  NEXT
```

This is because as soon as the asterisk appears, the rest of the line is passed to the operating system command line interpreter (OSCLI) and even if the NEXT were on a new line. OSCLI would not recognise the N% (this is peculiar to Basic and is not part of the

page 61 ▶

BBC **I.J.K.'s RANGE OF QUALITY SOFTWARE CONTINUES TO EXPAND...** BBC

MODEL A/B
Cassette 1: Star Trek/Candy Floss (very popular) **£6.50**
Cassette 2: Family Games (hours of fun) **£4.50**
Cassette 3: Mutant Invaders/Breakout **£6.50**
Cassette 8: Model A Invaders (M/C) **£5.50**

MODEL B (or A+32K)
Cassette 4: Beeb-Beep (Super Simon Game) **£4.50**
Cassette 5: Beebmunch (full colour Munchman) **£6.50**
Cassette 6: Super Hangman (animated, educational) **£4.50**
Cassette 7: 3D Maze (fast and intricate) **£4.50**

CASSETTE 9
Contains model B Invaders. A superb feature adaptation of the arcade 'Space Invaders' game in machine code and hi-resolution colour graphics for the BBC Micro model B (or A+32K). Play normal game or choose from the many options including Missile, Bomb and Invader speeds. Invisible Visible and Shields no Shields. Quite simply the best.
Only £7.50 for MODEL B (or A+32K)

CASSETTE 10
WORDPRO. Cassette based word processor for either Epson or Seikosha printers. Features right hand justification, alter, insert, delete, pages to tape, printer mode changes from within text line etc., etc. Complete with manual.
Only £10.50 inc. for MODEL B (or A+32K)

ALL OUR SOFTWARE IS AVAILABLE BEFORE WE ADVERTISE

CASSETTE 11
ATLANTIS. The superb fast action arcade game written in machine code to illustrate to the full the machines fantastic colour graphics and capabilities. This game includes all the usual ATLANTIS/SCRAMBLE features. Guide your submarine Nautilus along the undersea landscape and through the caverns avoiding mines, depth charges, rockets, jelly fish, serpants etc.
Only £7.50 inc. For MODEL B (or A+32K)

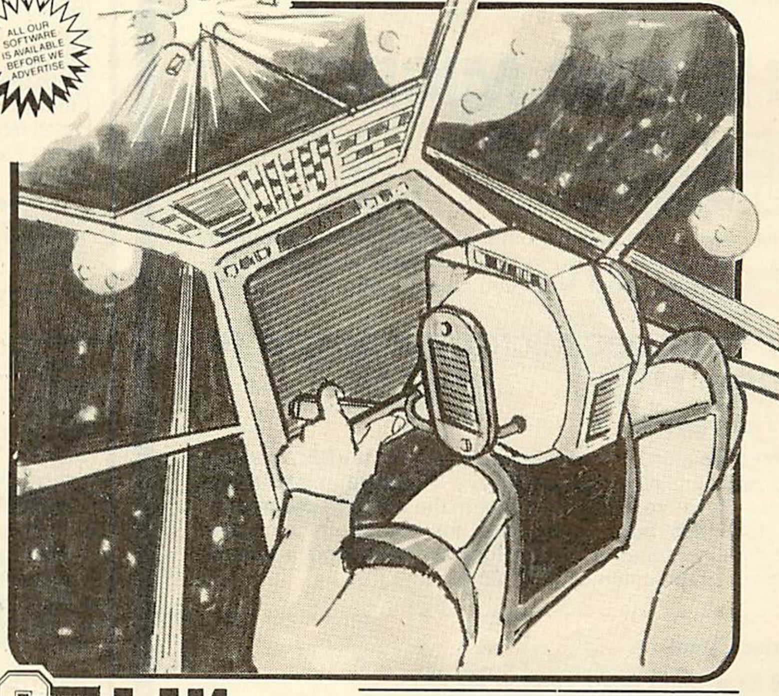
CASSETTE 12
FLAGS. A superb educational program. The flags of the world are drawn in hi-resolution colour graphics. The program then tests your knowledge of flags and geography. Have fun while learning.
Only £4.50 inc. For MODEL B (or A+32K)

BBC Micro Games

CASSETTE 13 – HYPERDRIVE
A new, very addictive machine code arcade game. Guide your laser tanks around the network of passages destroying the drone Aliens – but beware, evil OTTO lies in wait!
Only £6.50 inc. For MODEL B (or A+32K)

CASSETTE 14 – STRATOBOMBER
Another new highly colourful machine code arcade game. Can you keep the enemy fleet at bay long enought to destroy the nuclear reactor of the rogue Star Ship before it destroys your home planet? Superb graphics.
Only £7.50 For MODEL B (or A+32K)

Deduct £2.00 when ordering 3 or more cassettes



WE'RE EXPANDING TOO... NOTE OUR NEW OFFICE ADDRESS AND CUSTOMER SERVICES

ALL PRICES FULLY INCLUSIVE OF VAT AND P&P – NO MORE TO PAY!

- All Programs will run on all operating systems
- All software in stock before we advertise
- Send SAE for Brochure



9 King Street,
Blackpool, Lancashire





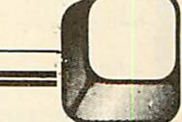
24 HOUR ANSAFONE

48 hour despatch

OFFICES OPEN
10am – 5pm Tues-Sat







BBC MICRO GAMES



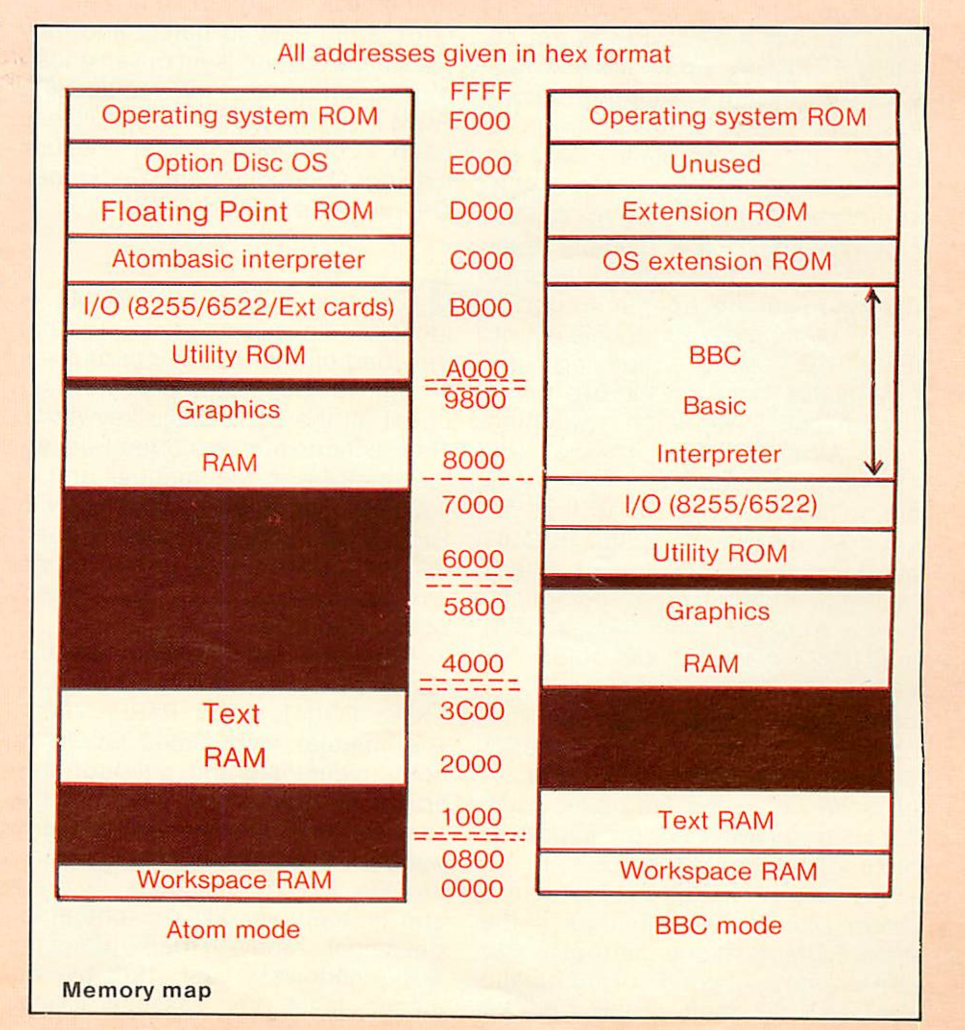
Running a better Basic has advantages, but Barry Pickles finds Acornsoft's conversion has several drawbacks. Next month he puts the board to use.

Atom owners, although they possess a versatile machine, tend to look enviously at their counterparts who are BBC micro owners. Because of this, and in line with their policy of 'upward compatibility', Acornsoft have recently issued an add-on board, providing an implementation of BBC Basic for Atom owners. Let me begin by stating, that an Atom is not – and can never be – a BBC micro. The Beeb is a very different beast and has far more features and expansion possibilities than could ever be provided on the Atom. However, for many users, this may not be important and the ability to use a wider range of software, or to have a more common Basic, may be all that is required. So, what do you get for your £50?

Opening the box revealed the board, wrapped in a paper-thin sheet of foam and accompanied by a slim booklet. The booklet turned out to be the 'BBC-type Basic Manual', an expression which was well chosen, as its even worse than the BBC Provisional User Guide! It must rank as one of the lousiest pieces of documentation ever, containing only 38 pages, with four example programs. Opening the booklet revealed that the board required the 6522 versatile interface adapter to be fitted and that a full 12k of RAM was preferable (it always is!). It also stated that you could not use this board with Econet, the colour board, or the Atom disc unit (this turned out to be incorrect).

Acorn say this board will work with the standard power supply, providing you use low power memory chips and have no colour board or external additions. That's true, but a bit like running a car flat out all the time and, as the board actually touches the heatsink, I would recommend that you upgrade to a 3A supply, as soon as you can.

After the rather depressing introduction, the booklet goes on to explain in great detail how to fit the board. It is plugged into four existing IC sockets on the Atom PCB and overlays all the existing text RAM area – and more. The board is a tight fit and you must



Memory map



align the pins exactly before pushing them home. This is not easy and, if you are in any doubt as to your ability to do ths, you should get your dealer (or someone at your local club) to fit the board – it's not worth the risk of damage. Once fitted, one link on the main PCB, and one flying lead to PL7 need to be soldered. If you care to solder three more wires to the PCB, this will allow you to select the language used from the keyboard – the alternative is to use the switch on the BBC board itself. If all is well, the usual display appears on power-up and pressing BREAK gives the greeting 'BBC BASIC'. Pressing CTRL and BREAK simultaneously will switch to AtomBasic, and pressing SHIFT and BREAK puts you back into BBC mode. Before discussing the language itself, a brief mention of how this board works.

The board contains the massive 16k Basic interpreter ROM, a 4k MOS (machine operating system) ROM, 2k of static RAM and some miscellaneous components. Additional sockets are provided for a 4k 'utility' EPROM and a 4k 'MOS extension' ROM – although Acorn have no plans to issue such a ROM. The 2k of static RAM is usable in Atom mode, where it is addressed from £2000 to £27FF. Figure 3 shows the 'memory map' for the two configurations and, in BBC mode, it has been arranged to be as compatible as possible with the BBC micro. Studying the relationship between the two maps will reveal that, when switching from Atom to BBC mode, the address lines for the text memory are shifted down by 8k so that, for example, memory at &0900 in BBC mode will be addressed at £2900 in Atom mode. This is potentially useful. Those of you familiar with the BBC machine will know its MOS uses 16k of ROM and it will be obvious that the 4k used on this board will not offer all of the BBC's features. This seems a good point to move on to the language itself, but, before doing so, an important announcement.

As announced in January's *Acorn User*, there is a bug in the MOS ROM! It affects all those who use a printer having an automatic linefeed. Normally, the printer routine should not send a linefeed character to the printer and so £FE contains this character (£OA), which is compared with each byte of data sent, and ignored by the printer driver, if a match is found. Here, however, the printer routine stores a different character at £FE with the result that the printer will only print with double line spacing. In fairness to Acorn, this kind of error is not easy to detect and they say that, if anyone is inconvenienced by this error, they will replace the ROM free of charge. The fault has been corrected on all issues leaving Acornsoft from December. OK, on to the language . . .

x value	Action
0	Normal listing
1	Single space after line number
2	FOR...NEXT loops indented to current depth of nesting
3	Actions 1, 2
4	REPEAT... UNTIL loops indented to current depth of nesting
6	Actions 2, 4
7	Actions 1, 2, 4

Table 1. LISTO (x) options

The 16k interpreter ROM is identical, in all respects, to that provided on the BBC micro and will recognise (but not necessarily act upon) all the BBC Basic keywords. Their actions are described briefly – too briefly! – in the 'manual' and, if you are not already familiar with BBC Basic, you will certainly need the BBC *User Guide*, or one of the 'Programming the BBC micro' books available.

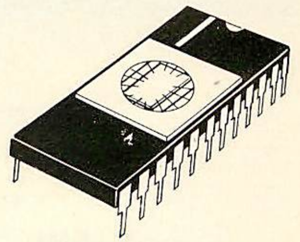
Keywords not implemented are: ADVAL; COLOUR; ENVELOPE;EOF; EXT£; POINT; POS; PTR£; VPOS. The manual also states GCOL is not implemented and, although this appears largely true, I do have a program that does not run properly without GCOL. Also not implemented are the PLOT commands to draw and fill triangles, VDU23 (character definition), and VDU28 (defining text 'windows'). Last, but by no means least, the cassette format retains Atom standard, which means you will not be able to load software tapes for the BBC micro. This must be considered a serious omission and I cannot understand why Acorn have left it out, since it would considerably increase the potential market, both for this board and their own software.

Space does not permit a description of all the available keywords, so here are those Atom owners will find delightful: READ;DATA;RESTORE.

CHAIN allows you to load and run a program, calling it from another program. EVAL evaluates an expression within a string. Useful for inputting formulae. LISTO provides optional methods of listing programs, none of which use anyu more memory. See table 1 for the options.

PRINT uses the variable @% to select a multitude of options, including proper numeric formatting. SOUND A 'beep' type command, with pitch and duration selectable. TIME provides a timer. It can be set within a program and will 'tick' at the rate of 1/100sec. It may be read at any time and is useful for providing a 'timeout' feature. DEFN defines a function, which can be any expression and use as many arguments (variables/constants) as you wish. These may be the current value of the argument or may be LOCAL to the function. DEFPROC defines a PROCedure. Definition is carried out in much the same way as a function, and, again, the arguments can be global or LOCAL. See Paul Beverley's article

page 46 ▶



Software Savers

WORD PROCESSOR PACKAGE

ONLY

£25.50

(including VAT and Delivery)

The most advanced word processing program written in machine code and permanently stored on a ROM which plugs into your BBC computer board, (Model B only). This program contains many advanced features including access to existing programs, and is undoubtedly the best value for money. If you are not convinced, we guarantee to refund your money and postal expenses.

BUSINESS PROGRAMS

Sales Ledger	£10.00
Filing	£10.00
Purchase Ledger	£10.00
Payroll	£10.00
Invoicing	£10.00
Home Accounts	£10.00
Stock Control	£10.00
Directory/enquiries	£10.00

GAMES

Backgammon	£10.00
Space Invaders	£7.00
Pucman	£7.00
Space City	£7.00
Gambling Video	£7.00
Star Trek	£10.00

FREE Program with any order over £45.00 in value
Send S.A.E. for catalogue. To order send cheque or phone us with your credit card number

All prices include VAT and delivery

SOFTWARE SAVERS

Temple House, 43-48 New Street, Birmingham B2 4LH

Telephone 021 643 4577 Telex 337045



	Floating point AtomBasic	Floating point BBC Basic	Integer AtomBasic	Integer BBC Basic
BM1	N/A	0.96	1.20	0.34
BM2	13.50	5.47	5.70	3.38
BM3	31.50	15.49	11.40	14.58
BM4	28.20	16.65	13.50	15.11
BM5	31.00	19.07	14.20	17.55
BM6	N/A	27.87	29.60	21.25
BM7	N/A	42.54	42.90	30.32
BM8	277.00	92.06	N/A	90.69

Table 2. PCW benchmark timings (seconds)

in December's *Acorn User* for details of these powerful commands.

There are also a number of keywords which provide useful toolkit and debugging facilities.

Names used for variables (string or numeric), functions or procedures can be of unlimited length and contain any mixture of alphanumeric characters, although the first one must be an alpha character. Keywords may not be used to begin a name, thus LISTOUT will cause an error but OUTLIST is OK. Finally, multidimensional arrays are allowed and, as in AtomBasic, these must be dimensioned first.

Ok, so how easy is the language to use? BBC Basic is close to Microsoft (widely used on many computers), which means that there is a large amount of printed software which will run without modification. As you get used to the language, you will find many of these programs can be re-written to take advantage of the advanced features of BBC Basic. Published games programs will require more thought, since many of the BBC's advanced graphics features are absent, but few of the problems are insurmountable. The graphics screen is presented as a standard screen of 1280x1024 points in all modes, automatic scaling being performed as you switch between modes, and many graphics routines will run without modification. For numeric printing, the print formatter is a dream!

I suspect many users will have problems using things which are common to the syntax of both languages, but produce a different result in each. First, multi-statement lines use the colon(:) as a statement delimiter. PRINT will print a new line unless you tell it not to, by using a semi-colon(;) – Atom does just the opposite! In BBC Basic, arithmetic is normally performed in f.point, unless you specify integer variables using the suffix % (eg TOTAL%) – again the opposite of AtomBasic. Strings still use the $ symbol, but this is now placed after the variable name. The hash symbol (#) is used only in assembler, to signify an immediate operand (@ in Atom), and hex numbers are now noted by prefixing them with the ampersand (&). Finally, although the manual doesn't tell you, the inverted up-arrow (↑) is used to PRINT a hex number. You'll get used to it after a while, but it helps to have two heads!

BBC Basic is fast, up to 3.5 times faster than AtomBasic. Standard 'benchmark' timings are given in table 2. Since the Atom runs at 1MHz, BBC programs will run at about half the speed they would on the real thing. If this bothers you, the National Atom Users Group published a 2Mhz modification in their newsletter no 6.

With the board fitted, you have a configuration not unlike that of the BBC model A, but any comparison is unfair.

I have two criticisms of this package. Fist, the documentation is not good enough. Acorn should have included the proper BBC *User Guide*, or, at least the relevant parts of it. The second is the lack of a BBC cassette standard and some of the most common OS routines. All of the important facilities could have been in a second ROM and, whilst there is room here for enterprise, the trouble is that no independent house is going to market such a ROM until it's sure enough boards have been sold to give an economic market base. On the other hand, the lack of these facilities may deter users from buying the board – Catch 22! Any move must come from Acorn – so how about it, lads?

This aside, the board is more useful than may appear and, having used it for two months, I find its features more and more appealing. The language is fast, powerful and fairly portable, and there is an enormous amount of printed software available. If your interest lies in games, you may have to do some work but, for other applications, BBC Basic offers just about everything that you could wish for. As I've said before, the Atom is not a BBC micro but, nonetheless, programmers who wish to develop software for the Beeb could do so cheaply, with this board.

In short, providing the limitations do not rule it out completely, the BBC conversion card is useful, and reasonable value at £49.95. The board is available from Acorn dealers or, by mail order from Acornsoft.



ERROR HANDLING

Debugging can be frustrating because of the lack of detail from Atom error codes.

Tony Armitstead has developed his own way

Atom users know how frustrating it can be to debug a program. The computer gives you the line number on which the error occurred and its code number, it is then necessary to look up the error code in the handbook. A better error handler would print out the statement in which the error occurred, a description of the error and then the error code and line number. The assembler program listed here provides such an error handler for the Atom at the cost of mode 4 graphics.

The error messages used can be the standard Acorn ones or better versions. The only restriction is that their total length in bytes is less than 914, and a lot of messages will fit in that space.

To use the program, type it in and check carefully before trying RUN, as an error could be disasterous. A hex number is printed out and it should be #9400; if not re-check the program. Turn to the error codes at the back of the manual and go through, entering the error codes and new messages. There is no need to bother with error 248 (it cannot be generated from within a program) or any other code you wish to omit. After entering the last code (238) you must enter code 255 with a message to the effect "UNLISTED ERROR". A hex number is now printed out indicating where the list ended and this must be noted. Set up your tape recorder, and using the number just printed out, type

SAVE "ERROR HANDLER" 9400 (list end)

The program should also be saved using

SAVE "ERROR PROG"

The code should now be tested directly by typing



```
1Ø DIMLL2,A4Ø;LLØ=-1;LL1=-1;LL2=-1
2Ø FOR J=1TO 2;P=#94ØØ;P.$21
3Ø T=#8Ø;S=#81;W=#FFED;X=#FFF4
4Ø[:LL2\ Set up routine
5Ø LDA @#ØB;STA #2Ø2
6Ø LDA @#94;STA #2Ø3
7Ø RTS
8Ø\ Store error number at Ø
9Ø PLA;TAX;PLA;STA Ø;PHA;TXA;PHA
1ØØ\ If graphics clear screen
11Ø LDA #BØØØ;BEQ P+7;LDA @12;JSR X
12Ø JSR W   CR/LF
13Ø\ Print statement
14Ø LDY @Ø;INY;LDA(#5),Y;JSR X
15Ø CMP @#D;BEQ P+6  CR found
16Ø CMP @#3B;BNE P-12 Semicolon found
17Ø JSR W CR/LF
18Ø LDA @#6E;STA T;LDA @#94;STA S
19Ø:LLØ\ Find error message
2ØØ LDY @Ø;LDA(T),Y;CMP@#FF
21Ø BEQ LL1 No message listed
22Ø CMP Ø;BEQ LL1 Error no. found
23Ø\ Step over error message
24Ø LDA @#D;INY;CMP(T),Y;BNE P-3
25Ø INY;TYA
26Ø\ Add y to table pointers
27Ø CLC;ADC T;STA T;BCC P+8
28Ø LDA S;ADC @Ø;STA S
29Ø JMP LLØ
3ØØ:LL1\ Print error message
31Ø INY;LDA(T),Y;JSR X;CMP @#D
32Ø BNE LL1 Upto cr
33Ø\ Jump to normal error handler
34Ø JMP #C9D8
35Ø]
36Ø NEXT J;P.$6
37Ø P=#946E;@=Ø
38Ø P."#"&LL2'
39Ø DO
4ØØ  INPUT"ERROR NUMBER"N
41Ø  ?P=N;P=P+1
42Ø  INPUT"ERROR MESSAGE"$A
43Ø  $P=$A
44Ø  P=P+LEN(P)+1
45Ø UNTIL N=255
46Ø PRINT"LIST ENDED AT #"&P
47Ø END
```

Error-handling program

LINK#9400: SILLY ERROR and return.

The Atom should reply

SILLY ERROR (your error message for error 94)
ERROR 94

If the Atom crashes or the screen fills with snow, there is a bug in the program. If all goes well the error handler is ready to test and use. Type NEW and enter a small program, which must have LINK#9400 as its first line, and an error in it. When RUN it should give the erroneous statement printed out, together with an error message, an error code and the line number.

The code and table are not affected by BREAK, but would be erased by CLEAR 4. If using graphics, develop programs in mode 3, and when it works, change to mode 4. If an error occurs whilst in a graphics mode, the screen is cleared before printing starts.

A word of warning; if using machine code and BRK is programmed, the error message produced is meaningless but you should realise this if you know what BRK does!

How does the program work? When entered, error codes and message are stored directly after the machine code, as in figure 1.

Figure 1.

ERROR CODE	ERROR MESSAGE	#ØD

```
 #9400
ERROR NUMBER?94
ERROR MESSAGE?UNKNOWN COMMAND
ERROR NUMBER?255
ERROR MESSAGE?UNLISTED ERROR
LIST ENDED AT #9490>
>LINK#9400;SILLY ERROR

SILLY ERROR

UNKNOWN COMMAND

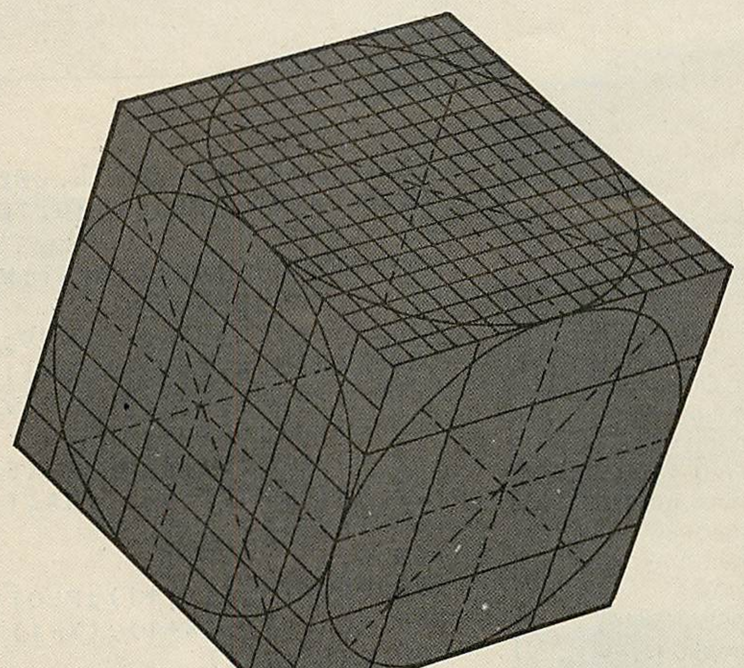
ERROR 94
```

Sample run

LINK#9400 resets some internal pointers so that when an error occurs control is passed to the new error handler. The error code has already been placed on the stack, so we extract this, put it in its correct place (at location 0) and restore the stack. It then prints the offending statement which is pointed to by locations #6 and #5. We now find the correct error code in the table and print the error message which is terminated by 0D. Finally, control is passed to the normal error handler.

A point to notice is that if there were many valid error codes we would enter them in the order of most probable occurrence, to minimise the time taken looking up errors. The maximum number of valid error codes for the Atom is only 37 and since m/c has been used the above remark is insignificant.

The time taken to load the code and table is about 20 seconds, and its ease of use means it can be used at every programming session to reduce program development time.



WIRE FRAME GRAPHICS

Three-dimensional wire frame objects can be built up using mode 4 in a 12k Atom. Philip Tubb uses a cursor to set up the graphics and rotate the figure around one or two axes simultaneously

Rotation program

```
5 DIM B(8Ø),C(8Ø),D(8Ø),E(8Ø)
6 DIM F(8Ø),G(8Ø),W(3),RR(2),P(-1)
7 V=1;X=99;Y=15Ø;L=1;@=1ØØ;J=2
1Ø GOS.47Ø;P.$21
3Ø[
31:RRØ JSR #FE71;STY#8Ø;RTS;]
33 P.$6
35 CLEAR 4
36 MOVE128,136;DRAW128,152
37 MOVE128,4Ø;DRAW128,56
38 MOVE12Ø,144;DRAW 136,144
39 MOVE12Ø,48;DRAW 136,48
4Ø REM keyboard  input
5ØsLINK RRØ;A=?#8Ø
6Ø IF A=42;Y=Y+1;G.a
7Ø IF A=34;X=X+1;G.a
75 IF A=46;Y=Y-1;G.a
8Ø IF A=54;X=X-1;G.a
81 IF J=1 OR A<>38;GOTO 88
82 P.$7
83 IF L=2;D?V=Y ;J=1;G.s
84 B?V=X ;C?V=Y ;J=1;G.s
88 IF J=2;G.91
9Ø IF A=52;J=2;G.b
```

page 50 ▶



This 3D rotation program aims to provide a comprehensive visual aid to designers and technical drawing tutors. Input can come from a taped file or from describing an object using a moveable cursor. Once this is done, a high resolution wire frame representation of a third (end) view is displayed followed by an isometric view and rotations around either two axes simultaneously, or one axis. Finally, an object may be recorded on tape for later use.

The program uses Atom mode 4, and makes use of trig functions, so 12k RAM is needed.

If, at the start of the run, it is decided to enter a new object, press D as instructed. Two crosswires will appear on the screen. Around the top crosswire, a side view of the shape is drawn. This is done by moving the flashing cursor using the keys: V=left, B=right, J=up, N=down. Only straight lines may be drawn, and this is done by showing the computer where the two ends of each line are. Position the cursor where the line is to go from and press F; then move the cursor to where the line is to go to and press T. Once the side view is completed press Q and the cursor will jump to the lower crosswire; here the plan view is to be drawn.

The task is much easier now because the computer is only looking for the distance of each line up the screen and will find the x distance itself. Now we come to the golden rule of drawing. In each view *every* line (or edge) on the object must be drawn, and *all* lines in the second view must be drawn in the same order as on the first.

Once both views have been described, the end view is displayed by pressing any key; if RETURN is pressed, the isometric view is given. After this, RETURN will rotate the object about the x and y axes simultaneously. For rotation about the x axis only, press X and RETURN. When satisfied with the object, press R and RETURN. Instructions for recording the object will then be shown.

from page 49

```
91 IF @=V AND J=2;G.200
92 IF A<>49 OR J=1;G.100
93 P.$7
94 FOR X=1 TO 10;WAIT;WAIT;N.X
95 IF L=1;@=V
96 IF L=1;L=2;V=1;X=99;Y=48;G.s
100PLOT13,X,Y;WAIT;WAIT;PLOT 15,X,Y;G.s
120b IF L=2;G?V=Y;A=D?V;G.130
123 E?V=X;F?V=Y;A=C?V
130 MOVE(B?V),A;DRAW (E?V),Y
131 WAIT;WAIT;WAIT;WAIT;WAIT
135 V=V+1;P.$7
140 G.s
150aPLOT13,X,(Y+1);PLOT13,(X-1),(Y-1)
155 PLOT 13,(X+1),(Y-1)
160 WAIT;WAIT
170PLOT15,X,(Y+1);PLOT15,(X-1),(Y-1)
175 PLOT 15,(X+1),(Y-1);G.s
200 GOS.n
204 REM iii end elevation iii
205 CLEAR4
210 F.A=1 TO V-1
215 MOVE((D?A)+90),((C?A)-48)
220 DRAW((G?A)+90),((F?A)-48)
225 N.A
230 REM ii rotation routine ii
250 INPUT $W;%N=0.25;L=1
251 %N=%N+0.43
260 CLEAR4
270 FOR A=1 TO V-1
280 Z=(D?A)-48;X=(B?A)-128
285 Y=(C?A)-144;GOS.c
290 MOVE (128+K),(96+Q)
300 Z=(G?A)-48;Y=(F?A)-144
305 X=(E?A)-128;GOS.c
310 DRAW (128+K),(96+Q)
350 NEXT A
352 INPUT $W
354 IF $W="" ;G.251
357 IF $W="X";L=2;G.251
360 IF $W="R";GOTO e
361 END
370 REMiiiii rotating iiiii
380c IF L=2;%Z=Z;K=X;G.y
382 %T=ATN(Z/X);%I=Z/SIN(%T)
385 %K=COS(%N+%T)*%I
387 %Z=SIN(%N+%T)*%I;K=%K
390y %T=ATN(Y/%Z);%I=Y/SIN(%T)
393 %L=COS(%N+%T)*%I;Q=-(%L);R.
470 P.$12
471 REM the start of the prog
475 P."          ORTHOGRAPHIC TO"'
477 P."          *3D* CONVERTER"''
480 PRINT"    choose mode"'
```

page 51 ▶

from page 50

```
485 P."'D' DESCRIBE A NEW OBJECT"'
490 INPUT"'L' LOAD A  STORED  FILE"$W
500 IF $W="D";R.
505 REM tape file input
510 PRINT"    loading from tape"'
512 FOR J=1 TO 40;WAIT;WAIT;N.J
515PRINT"POSITION TAPE ON FILE LEADER"'
520 PRINT"play AND PRESS A KEY"'
525 LINK #FFE3
530 V=BGET X
535 PRINT"file found"'
540 FOR Y=1 TO V
550 B?Y=BGET X;C?Y=BGET X;D?Y=BGET X
555 E?Y=BGET X;F?Y=BGET X;G?Y=BGET X
560 NEXT Y
570 PRINT" loaded;PRESS A KEY"'
575 LINK #FFE3
577 CLEAR4
580 FOR Y=1 TO V
590 MOVE (B?Y),(C?Y);DRAW (E?Y),(F?Y)
595 MOVE (B?Y),(D?Y);DRAW (E?Y),(G?Y)
600 NEXT Y
602 V=V+1
605 INPUT $W;G.205
615 REM ii recording data ii
620e CLEAR 0
630 PRINT"    recording object"'
632 F.J=1 TO 40;WAIT;WAIT;N.J
634 P."IF YOU DON'T WANT SHAPE RECORDED"
635 P."JUST PRESS A KEY"''
637 P."PUT TAPE INTO record AND ALLOW"'
640 P."A LONG LEADER THEN PRESS A KEY"'
645 V=V-1
650 LINK #FFE3
660 BPUT X,V
665 WAIT;WAIT;WAIT;WAIT;WAIT
670 FOR Y=1 TO V
680 WAIT;WAIT;WAIT;BPUT X,(B?Y)
685 WAIT;BPUT X,(C?Y);WAIT;BPUT X,(D?Y)
690 WAIT;BPUT X,(E?Y);WAIT;BPUT X,(F?Y)
695 WAIT;BPUT X,(G?Y)
700 NEXT Y
710 PRINT" file recorded"'
720 PRINT"    PROGRAM FINISHED";END
730 REM iiii bug killer iiii
735n F.A=1 TO V-1
740 IF D?A=48;D?A=49
745 IF B?A=128;B?A=129
750 IF G?A=48;G?A=49
755 IF E?A=128;E?A=129
760 IF C?A=144;C?A=145
765 IF F?A=144;F?A=145
770 N.A;R.
```

C. J. E. Microcomputers

BBC Specialists

VAT Included where applicable

Quality Disk Drives

Single drive	40 track single sided 1 × 100k		£200.00
Dual drive	40 track single sided 2 × 100k		£350.00
Dual drive	40 track double sided 2 × 200k		£546.25
Dual drive	80 track double sided 2 × 400k		£799.25

All drives are cased with own PSU for reliability, and include connecting cables and utilities disk. Delivery £4.00

Software for the BBC Micro

MISSILE CONTROL the first implementation on the BBC Micro of the popular arcade game. (32K)	£9.00
MAZE MAN an authentic version of the popular arcade game. (32K)	£6.00
BALLOONS a highly original game that soon becomes compulsive playing. (32K)	£6.00
DISSASSEMBLER the memory dump routine includes a scrolling back in memory facility. (16K)	£5.00

MISSILE CONTROL, MAZE MAN and BALLOONS use the Keyboard or Joysticks for control

30+ Programs for the BBC Microcomputer

This Book contains program listings, with explanations and tips on using the BBC Micro
GAMES UTILITIES GRAPHICS and MUSIC
Most programs will run on Model's A and B
Edited by C. J. Evans, various Authors.
A pair of cassettes with all the programs is available.

BOOK £5.00
BOOK and CASSETTE SET £9.00

LEADS

The BBC Micro comes without a cassette lead

7Pin Din to 2 × 3.5mm and 1 × 2.5mm minijacks	£4.00
7Pin Din to 5Pin Din and 2.5mm minijack	£4.00
7Pin Din to 7Pin Din	£4.00
7Pin Din PLUGS	Two for £0.65
6Pin Din PLUGS (for RGB socket)	Two for £0.65
5Pin Din PLUGS (360' for RS232)	Two for £0.65

RS423 TO RS423 (BBC Micro to BBC Micro)
Two metre cable £4.00 Four metre cable £5.00

TELEVISION/MONITOR LEADS full range available

Phono plug to Co-ax with high quality cable 3 Metres	£3.00
BNC Plug to BNC Plug	£3.10
BNC Plug to Phone plug (i.e. BBC Micro to Rediffusion TVRM)	£2.20
RGB 6PIN DIN to 6PIN DIN	1 metre £4.00 2 metre £5.00

PRINTER CABLES

BBC to 36 way Centronics Type connector	£17.50
BBC to 25 way D Type (For use with RS232)	£ 9.50
BBC to 40 way edge connector (Centronics 739)	£20.00
TORCH to 36 way Centronics Type connector	£20.00

Blank C30 Computer Cassettes

Ten for £4.50
15 Way D Type Plug with Cover £2.75
Computer graphics design pads 100 sheets £4.00

BBC Upgrade Kits

RAM UPGRADE (100ns)	£23.00
KIT A Printer and I/O Port	£ 9.50
KIT B Analogue Port	£ 8.00
KIT C Serial I/O and RGB	£10.00
KIT D Expansion Bus/Tube	£ 7.50
Full Upgrade kit	£58.00

All components full specification

STAR DP8480 PRINTER From £250.00 Inc VAT

80 CPS : 80/96/132 COLS
BIDIRECTIONAL LOGIC SEEKING
TRACTOR WITH FRICTION FEED

CENTRONICS £217.39 + £32.61 VAT = £250.00
RS232 £235.00 + £35.25 VAT = £270.25

High Res Graphics option to allow BBC Screen dumps £15.00/£20.00
(24HR SECURICOR DELIVERY FOR PRINTERS £8.00)

VAT Included where applicable

Send SAE for full Price List of our large range of accessories.

C. J. E. Microcomputers

POSTAGE Add 50p per order or as stated

Dept (AU), 25 HENRY AVE, RUSTINGTON, W.SUSSEX. BN16 2PA (09062) 6647



LANSDOWNE

ACORN ATOM

17K RAM BOARD—Holds 1K × 2114 and 16K × 6116 Cmos Ram. Memory mapped between 3C00 and 7FFF. Fitted with Edge Connector for internal fixture to PL7. NO EXTRAS NEEDED.

Min, Config. 1K.............£31.00
Full 17K.............£64.00

5 VOLT 3 AMP POWER SUPPLIES
Enclosed power supplies for the expanded Atom. Complete with all leads.........................£28.75

ALL PRICES INCLUSIVE

AVAILABLE END FEB—MULTI BOARD
Holds up to 8 memory devices, which may be any of the following:
25/27 series Eproms, 5116/6116 Cmos Rams or 4802 Static Ram. Memory map user selected. Write for details SUITABLE FOR RACK OR ATOM USE.

LANSDOWNE ELECTRONIC SYSTEMS
25 Lansdowne Street, Darlington,
Co. Durham DL3 0NJ Tel: (0325) 486000



BRANCHING OUT

Simon Dally becomes a tree surgeon for this month's brainteaser. Acornsoft programs worth £20 await the three lucky winners.

Ten years ago at university, believing it wouldn't happen in my lifetime, I laid a bet with a fellow student that no computer would achieve chess grandmaster status within 20 years. My reasoning was based on the then popular assumption that the number of permutations in a game of chess was so enormous as to be, for practical purposes, infinite. For instance, it is probably a conservative estimate to say, that in *a random position*, in an average game of chess, there are likely to be four reasonable-looking moves available to the player whose turn it is to move, and four reasonable replies to each of those four moves.

A quick check with your calculator or computer will show that, looking ahead on this principle, a mere five moves (ten half-moves) – a move in chess constitutes a half-move from each player – leads to over a million different positions. Moreover, since a computer lacks the human ability to spot instinctively a reasonable move, .it has to examine every possible combination. Even if the computer could analyse a million moves a second, it would still take it several days to think its way through to a situation that a good club player can arrive at in a minute or two. This method of looking at a problem is usually referred to as a tree-search because you start on one trunk, then go down each of its branches and then each of the offshoots from the branches.

Since I made my bet computerised chess has made impressive advances. High-street shops now sell dedicated chess-playing machines, while software houses market programs for micros at a few pounds. I still fancy I have a good chance of winning my bet, but I'm now much less confident than when I made it.

The reason for the improvement in chess-playing programs is quite simple: the advances made in the algorithms which make up the programs, and in particular what traditional jargon refers to as 'tree-pruning'. In other words, rather than test every possible position, the computer embarks on one branch of the tree and scrambles back to the trunk if it finds it has previously gone down a more promising-looking branch. Human beings are very good at rejecting unpromising branches instinctively.

Chess continues to fascinate computer experts because, in theory, it should be possible to construct a program which will be world champion – though that day still seems a long way off.

Another tree-searching task which intrigues mathematicians and programmers, not least because it has considerable real-life applications, is the notorious travelling salesman problem.

Imagine you are a magazine distributor in a Manchester suburb, servicing six different corner-shop newsagents by providing copies of *Acorn User* and other goodies. Wear and tear on your van, petrol and labour costs obviously dictate that the route you take on your delivery round is the shortest and most economical.

Although you can probably work out the optimum route quite easily in your head it may surprise you to realise that there are 720 possible routes, ie 1x2x3x4x5x6, otherwise known as six factorial (6!). This is an even faster growing array of permutations than the chess game. And if you double your little empire to service 12 shops, you have nearly 40 million routes to choose from. With 15 outlets you have a staggering 110 billion possibilities.

In practice, things aren't quite as terrible as they seem. The same subconscious mental process which seems to reject the vast number of silly moves over a chessboard also seems to operate when it comes to the travelling salesman problem. In real life, factors like shop opening times may influence the choice of best route. However, it's hardly surprising, given these figures, that breweries, canned food manufacturers and other firms who deliver goods from city to city spend a lot of money employing people to tell them the best routes to take.

To my knowledge, there is no known algorithm which can find, in a practical amount of time, the best route for any number of places. However, I do know that a program containing our friend tree pruning can save a lot of time in determining the best route, when the number of places to visit is fairly modest.

Suppose we have to start at city A and visit six cities BCDEF and G before returning to A – and that the best route the computer has so far found is ABCDEFGA – a distance of 1000 miles. It now has to generate a new route, say, AEBCDEFGA. Suppose the computer calculates that the three-city route, AEBC and back to A, is greater than 1000 miles, then there is no point in continuing along this particular branch of the tree. (If you're puzzled about adding the distance to A at this stage the explanation is this: you'll have to return home in due course and the shortest distance between two points is a straight line, so you can ignore the intermediate cities while checking whether or not your current route is longer than the best found so far.)

The most important principle behind this discovery is that the machine can now discard the five

	BIR	BRN	BRL	CAR	COV	EDI	IPS	LEE	LON	LIV	MAN	OXF
BIRMINGHAM	-	176	84	193	18	291	163	112	121	99	86	63
BRIGHTON	176	-	150	364	154	452	132	250	54	270	257	99
BRISTOL	84	150	-	273	91	371	198	194	125	179	166	69
CARLISLE	193	364	273	-	211	98	308	122	309	124	117	258
COVENTRY	18	154	91	211	-	319	141	116	99	117	104	50
EDINBURGH	291	452	371	98	319	-	393	207	390	222	215	361
IPSWICH	163	132	198	308	141	393	-	198	73	254	226	129
LEEDS	112	250	194	122	116	207	198	-	194	70	42	158
LONDON	121	54	125	309	99	390	73	194	-	215	202	58
LIVERPOOL	99	270	179	124	117	222	254	70	215	-	36	164
MANCHESTER	86	257	166	117	104	215	226	42	202	36	-	151
OXFORD	63	99	69	258	50	361	129	158	58	164	151	-

Mileage table

remaining unevaluated permutations beginning AEBC.

More subtle forms of tree pruning can produce an even greater discard rate, so that a 12-city problem (40 million permutations approximately) can be solved by examining only a few thousand routes.

This month's competition features a travelling salesman problem and comes in two parts. The first is exclusively for children under the age of 13 – and you must give the name and address of your school and class with your answer.

1) Starting in London, you must visit each of the following cities once only and then return to London: Birmingham, Brighton, Bristol, Carlisle, Coventry, Edinburgh. What is the shortest possible route and how many miles do you have to travel? (The mileage chart is shown in table 1)

2) As in the first problem you start and finish in London. In addition to the cities mentioned in problem 1, you also have to visit once only Ipswich, Leeds, Liverpool, Manchester and Oxford.

a) What is the shortest route and how long is it in miles?

b) What does the most inefficient salesman do – ie what is the longest possible route visiting each city once only.

Answers on a postcard to February Competition, Acorn User, 53 Bedford Square, London WC1B 3DZ to arrive no later than 28 February 1983. Please also state which, if any, microcomputer you have.

NOVEMBER WINNERS

Our November competition brought in our biggest response yet – over 3000 entries.

The problem was first to find a nine-digit number containing all digits from 1 to 9 whereby knocking off the digits one by one from the right left a number exactly divisible by the amount of remaining digits. The second problem was to discover whether or not the puzzle editor was correct in declaring this number to be unique.

This problem was (I believe) first posed in *Scientific American* five years ago, and it cropped up again in the 'Brainteaser' column of the *Sunday Times* last summer. In both cases numerous readers claimed several numbers fitted the bill.

The answer is that the puzzle editor was correct! There is only one solution – 381,654,729. About 85% of entrants got this right. Those who got it wrong mainly came up with numbers such as 921,654,387: this is because dividing eight-digit numbers by eight on an eight-digit calculator display, and on certain micro-computers, is an inaccurate business.

Several people sent in multiple entries (presumably because they thought it would increase their chances of winning). So no marks to the person from Lancaster who submitted no less than 20(!) postcards in the same handwriting but with different names and addresses. Only the Post Office benefitted as the answer given – that there were no less than 9×10^8 solutions – far outstrips the number of permutations for the nine digits (362,880).

In one respect only was your editor guilty of supplying a problem with multiple solutions. A few pedantic wretches pointed out that since I had not stipulated a positive integer the word 'number' could embrace negatives and decimals, such as –381.654729 and that the readers were therefore right. Looked at this way there are 20 possible solutions, all with the same digits in the same order. I had to admit these snivelling attempts to do me down into the winners' pool. Sob. Even puzzle editors can't win them all.

Because of the massive response the editor has decided to award some additional prizes as well as the BBC microcomputer.

The winners are:

Paul Chauveau of Kent, who wins a BBC model A micro.

Ellen Dobson of London and **Stephen Hotchkiss** from the West Midlands who both win some free software.



Wordwise is the first word processing chip for the BBC micro to reach the market place, although others are soon to follow. Jacquetta Megarry plugs it in and sees what you get for your money

Plug-in chip improves your word power

It is to the personal and educational user that firmware such as the Wordwise chip will most immediately appeal as it brings professional word processing in sight. By separating text entry from editing, formatting and printing, word processing removes concern about making errors and encourages perfectionism in content and presentation.

Wordwise is supplied as a ROM chip, so it cannot be accidentally erased or corrupted. Once installed, it is instantaneously available and occupies no user memory, so about 4500 words can be stored. (Wordwise is intended for the model B but will also work on an upgraded model A.) Using external memory, such as discs, capacity is effectively unlimited.

Wordwise makes good use of the Beeb's distinctive features. In particular, all entering and editing is done at 40 characters to the line, so a domestic TV is perfectly adequate. At the touch of a button, the effect of formatting commands is previewed at 80 characters per line. A colour TV or monitor highlights the embedded commands in colour, though even in monchrome they are distinguished from main text.

The 10 user-definable keys are exploited to give easy control over the main editing commands. For example, f0 toggles between insert and overwrite mode (current status displayed in the top line, along with a running word count and reminder of characters available). The cursor arrows do exactly what you would expect, and, combined with the shift key, they do what you might hope: shift ← takes you to the beginning of a line, shift ↑ to the top of the document.

Fitting the chip inside the machine is surprisingly easy, even for a novice, and it is accompanied by instructions and a cassette containing sample text.

Many of the commands have a mnemonic quality: 'IN5' indents by five spaces, 'JO' turns the justify facility on, 'LS3' produces triple line spacing, and so on. Commands like these can be used anywhere in the text and their effect can be previewed immediately without disturbing the cursor.

Even more impressive is the facility with which the user can jump between different parts of the system. The main menu offers eight options, with intelligible labels like 'load new text', 'search and replace' or 'preview text'. Pressing the Escape key once or twice always returns the user to this menu, so that within half and hour, a total novice can be operating the system.

Furthermore, exit from the system is effortless; just by typing *BASIC, you turn your word processor back into a model B running BBC Basic. (In fact, I have arranged the chips so the machine switches on in Wordwise.) Similarly, commands like *TAPE, and *DISC allow instantaneous switching between cassette and disc systems.

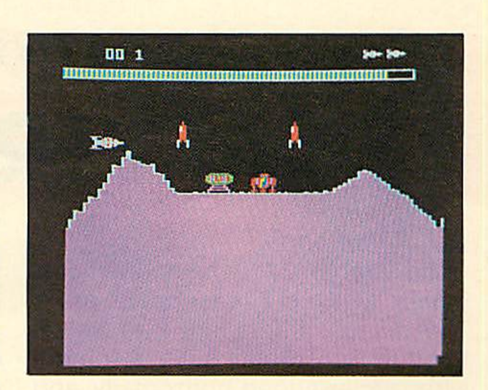
The editing power of Wordwise can be brought to bear on stored programs as well as text by these means, thus removing the tedium of debugging, revising and improving programs, and even of adapting them to run on other micros.

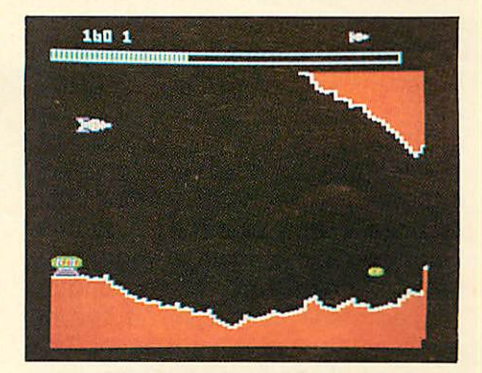
Wordwise supports fully automatic paging with running heads for up to 9999 pages. However, the user always feels in control; for example, making paging conditional can prevent a table being broken up by a new page. There are also precautions against absent-mindedness. The f0 key emits a short beep which prevents you from overwriting when you think you're inserting. If you want to delete text, you press f6 and are asked 'Delete to ?'. If you answer with a space, Wordwise obediently deletes the next word; if instead you reply with Return, it asks if you are sure before destroying the whole paragraph!

This system provides all the facilities a small business, school, self-employed professional, author or journalist could want. Furthermore, it is pervaded by thoughtful touches which suggest that the designer was familiar with the frustrations of first-generation word processing. For example, pad characters can be used to prevent



ROCKET RAID RAVE





THE lads at Acornsoft have done it again. Vidiots who derive immense pleasure and satisfaction from the likes of *Defender*, now have another favourite in *Rocket Raid*. It uses just as many keys (unless you have joysticks), but is even more difficult and is really about six games rolled into one.

Raid starts off as shown above right, with a *Defender*-type terrain, and lots of fuel dumps and other goodies to destroy, by bomb or laser. The only sources of peril are the rockets which are launched sporadically and the solid mountains.

Once through this landscape, you must negotiate the 'Tunnel of Love' and the green bouncing monsters.

After the tunnel come meteors, a sunken city, a maze and then your final target – a dustbin in a deserted city. Once you've bombed the bin, it's back to the start for a faster game.

During the later stages of the game, you have to get down a vertical wall, which is where the space bar and SHIFT keys come in handy. These don't appear to work properly in the early stages, but this is because they effect motion relative to the screen rather than the landscape, which is scrolled.

Rocket Raid runs on a model B and costs £9.95 from Acornsoft, 4a Market Hill, Cambridge, but you'll probably get it quicker from your local dealer, if past delivery delays are anything to go by.

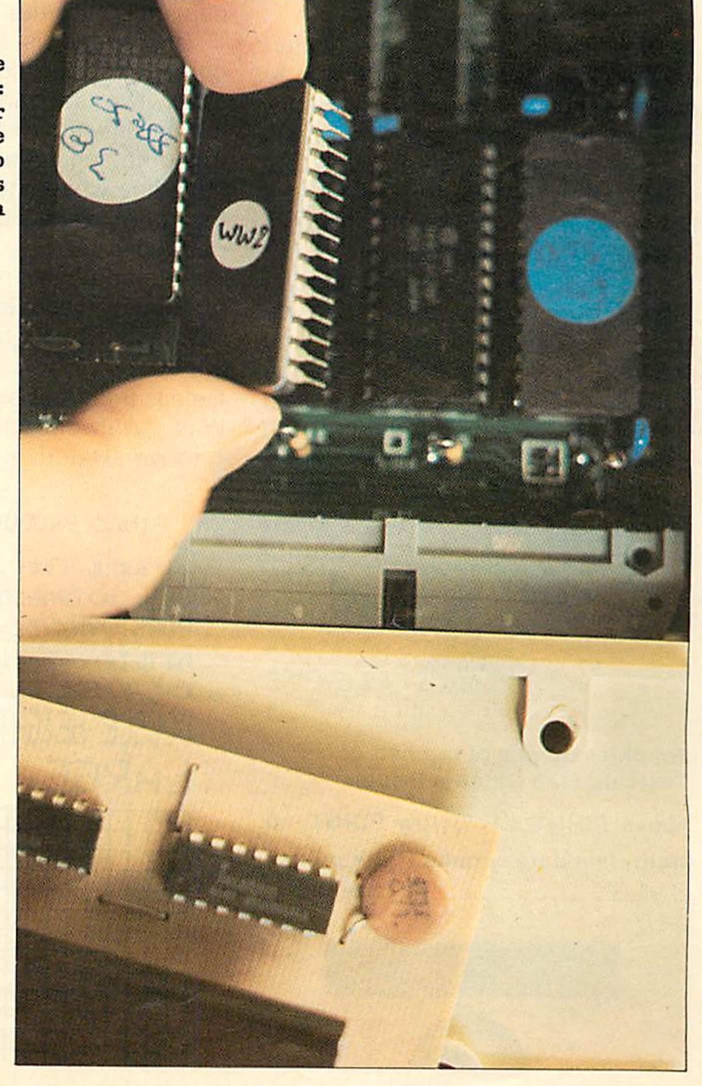
To sum up, a great game if you can get it!

Tony Quinn

justification of the right hand margin splitting a word awkwardly. Word counts and deletions can be executed on specified sections of text, not just on the whole document. Sections of text or program can be marked, searched, moved or copied. You can even program each of the function logs to produce useful strings, independenly of their specialised Wordwise functions. When doing letters, I reserve one for my letter-head, another for the signature, and so on; you can even call up a short standard paragraph.

Inevitably, there is a catch: Wordwise will not work with operating system 0.1. So you may have to buy a 1.0 ROM upgrade, or persuade Acorn to give you one. Computer Concepts, who distribute Wordwise, intend to supply customers with the chips at about £7.50 extra.

Wordwise will undoubtedly appeal to primary schools because of its cheapness, compactness and ease of use. It also helped this reviewer to report on a month's acquaintance with Wordwise without having to count her 1421 words (before editing).



Fitting the Wordwise chip: easy even for the novice thanks to the Beeb's design

● Wordwise is available from Computer Concepts, 16 Wayside, Chipperfield, Herts. It costs £39 plus VAT.



George Hill puts the £70 Amber 2400 to the test

FOREVER AMBER

The Amber 2400 is a dot-matrix printer at a very reasonable price. The manufacturers hope to sell it through Boots retail outlets. The dot wires are arranged horizontally, and produce a line of 144 dots. Each character is represented on a seven by five matrix. The paper width is 2.25 inches, and there are 24 characters per line.

Print quality is satisfactory (see examples), and the Amber can produce double-width, double-height, and double-width-and-height characters on sending single control characters. It can be used in either serial or parallel. This is accomplished by changing links on a circuit board and is easy to do (requires no soldering). The serial option has produced some peculiar results, however.

The Amber has a limited graphics capability. It is possible to address the dot wires individually, and so print pictures, but the small number of dots per line means that even a single-tone dump of the BBC screen needs to be done in two halves (see example).

Limitations of a 24 character width are readily appreciated if you type WIDTH 24, and list your program on the VDU. This limitation is further compounded by the fact that the BBC micro indents each line number as if it had five digits, so the first three characters are often spaces.

The instruction leaflet is quite helpful as it avoids the worst excesses of jargon and incomprehensibility, and contains sample graphics programs for TRS 80 and BBC micro. The BBC one only reproduces the bottom left hand quarter of the screen and is incredibly slow, taking 8 mins 45 secs for its quarter screen dump, and having the rather curious property of selecting COLOUR 1 to be printed, and ignoring all others.

Only one roll of paper is provided, and one ribbon, and there is no information on where to obtain replacements. The ribbon prints in an abominable purple, but is in a neat little cassette.

The speed is low. It prints 12 characters per second, or a line every two seconds. The graphics dumps illustrated took 5.5 minutes with the enclosed program, and have fine white lines on them where the print wires fail to marry up perfectly.

To sum up, it's a neat little printer producing reasonable quality output but limited by its narrow paper and speed.

Details from Amber Controls, Central Way, Walworth Industrial Estate, Andover, Hampshire.



Graphics dumps of top half of Testcard (see October issue)

Above POINT >1, below POINT >0.

Right, test data supplied with printer.



FINAL PRODUCTION TEST

!"#$%&'()*+,-./01234567

89:;<=>?@ABCDEFGHIJKLMNO

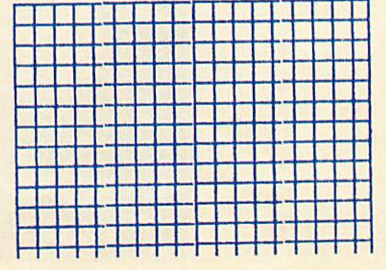
PQRSTUVWXYZ[\]↑_`abcdefg

hijklmnopqrstuvwxyz{|}~£

DOUBLE WIDTH

DOUBLE HEIGHT CHARACTERS

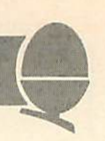
LARGE PRINT



```
 1000 REM * * AMBERPIC *
 *
 1001 REM * Copyright G.
B Hill November 1982 *
 1002 REM picture dump f
or use with BBC MICRO an
d AMBER 2400printer
 1003 REM single colour
version
 1004 REM ***variable de
claration***
 1005 REM X%,Y%,y% scree
n coordinates
 1006 REM Z% store for P
OINT data
 1007 REM set up printer
 1008 PROCPRINTER
 1009 REM scan screen an
d send data to printer
 1010 PROCSCAN
 1011 REM finishing rout
ine
 1012 PROCDONE
 1013 END
 1014 DEFPROCPRINTER
 1015 REM zero graphics
cursor
 1016 MOVE 0,0
 1017 REM select paralle
l mode
 1018 *FX 5,1
 1019 REM send linefeeds
 to clear paper
 1020 VDU1,10,1,10,1,10

 1021
 1022 ENDPROC
 1023 DEFPROCSCAN
 1024 FOR A=0 TO 1
 1025 FOR X%=0 TO 1279 S
TEP 4
 1026 REM set graphics m
ode
 1027 VDU1,17
 1028 FOR Y%=A*512 TO 51
1+A*512 STEP 32
 1029 Z%=0
 1030 FOR y%=0 TO 31 STE
P 4
 1031 Z%=Z%*2
 1032 IF POINT(X%,Y%+y%)
 >1 THEN Z%=Z%+1
 1033 NEXT
 1034 VDU1,Z%
 1035 NEXT
 1036 REM finish line
 1037 VDU1,0,1,0
 1038 NEXT
 1039 VDU1,10,1,10,1,10
 1040 NEXT
 1041 ENDPROC
 1042 DEFPROCDONE
 1043 REM reposition tex
t cursor
 1044 VDU4,26,31,12,21
 1045 PRINT;"Picture com
plete";
 1046 VDU31,18,24
 1047 PRINT;"BYE"
 1048 VDU31,0,29
 1049 ENDPROC
```

Graphics dump printed by Amber



Can static be blowing the amp?

Sir, We are a Norwegian group using the BBC model B in education. In this field we find the software part of the machine most excellent and sophisticated to use.

However, we are disappointed regarding the construction of the computer. The printed boards are not secured well enough for the bumps and shocks which are an everyday experience in a school. The micro has to be transported from classroom to classroom as another audio-visual tool and must withstand the stresses such use will develop. We have already soldered one of the printed circuits.

And what's worse, we have crashed five out of ten of our machines on cassette loading. It is the LM324 quad operational amplifier that's being destroyed. Why does this happen?

We know the cassette recorder has been connected and disconnected to the BBC while both were switched on. This could create static outbursts of destroyable voltage, but we don't think it is the reason. Anyhow, the micro must withstand use like this. We cannot rely on strict handling routines for cassette operating in a school where many different people handle the computer.

Has anyone reported this problem before? And if so, what is the reason for the destruction of the LM324?

We hope you can give us a clue.

Bernt Pedersen
Daisy Project
Norway

Well, we don't really know what to suggest for this one. No-one here or at Acorn has reported blowing up the LM324 like this.

Some people have crashed the machine and lost programs when unplugging the cassette, but nothing worse than that. This is usually cured by using a power supply for the BBC micro which is separate from that used for peripherals such as cassette recorders.

We can only suggest your wiring is incorrect or that it is the static causing the problem.

As you are probably aware, static discharge can be a major problem for computers.

Certain types of carpets, especially in dry, air-conditioned buildings can produce charges which will easily destroy tapes and some electronic components.

Sorry not to be of more help. Have any readers come across this? If so, information would be appreciated.

Article bug

Sir, I sat down to study my article 'Atom to Beeb' in January's *Acorn User* and thought 'What if. . . ?' Horror! There's a bug in it!

The line number of the program to be taped is stored as two bytes in the Atom. The problem arises if one of these two bytes is 13. This may not happen very often – it has never occurred when I use the program or I might have detected the 'bug' before.

The line will be taped correctly but then an extra line will be generated. The problem is that this line could overwrite a line when the program is loaded into the Beeb.

However, to overcome this only two lines need be changed. Line 4030 becomes:

```
4030$B=$(D+2); B=B+1+LENB;
    D=D+3+LEN(D+2)
```

Line 4040 becomes:

```
4040 U.B+LEN(D+2)> G+244
    OR ?D=255
```

Line 6010 may also cause confusion to some Atom owners. The last but one character of the line is represented as ¦, ASCII code 7C hex. It is obtained by shifting\. On the Atom screen it appears as an inverted \ and in the manual it is printed as ⍂. It represents the operation 'OR'.

C.J. Hollyman
Hants.

Double-speed chips

Sir, I received my BBC model B micro in July. I have sent my guarantee card off but have as yet received no reply.

I am having some problems with my computer. First and most important, I am unable to save/load programs. I have tried several different leads, different recorders, different tapes, and have taken care to ensure the volumes were correctly set. I have also tried the fix program given in *Acorn User* issue 2, all to no avail. Second, occasionally the computer will repeat a character very quickly so that even if I press on a key for a very short time the character is repeated.

J. Ingle
Scotland

The major problem with the computer is probably caused by the 6502 microprocessor being a double-speed chip. Acorn has been supplied with these incorrect chips occassionally, and this is usually brought to notice by the very fast key repeat.

Your dealer will be able to replace this chip.

If you have done all the things you mention, we don't understand why you can't load or save. Ask your dealer to check this when the chip is replaced.

Shuddering TVs

Sir, I recently added 16k RAM to my model A BBC micro as described in your magazine. However, in the model B modes (0–3) I do not get a steady picture on the television I am using for my micro, although everything is OK on two others. This particular colour television is about eight years old.

C.R. Dickens
Wiltshire

It sounds as if the colour guns in your TV have become misaligned. If the two newer sets work fine, it is unlikely to be the Beeb. You could try typing *TV0,1 which may reduce the shudder.

The lost chords

Sir, I want to improve the sound on my Beeb, but cannot find the audio out socket you mention in October's *Acorn User*.

L. Raynor
Sheffield

There are two audio connections to the BBC micro. The loudspeaker on the keyboard can be replaced by a larger one and the miniature rotary potentiometer turned to increase the volume. Alternatively the unamplified output of PL16 on the left hand edge of the PCB can be put into the tape input of an amplifier.



operating system). The alternative would be:

```
A% = &91 : X% = &60
FOR Y% = 0 TO 255 : CALL&FFF4 : NEXT
```

In assembly language this would look something like:

```
LDA #&91
LDX #&60
LDY #0
.loop
JSR &FFF4
INY
BNE loop
RTS
```

When you want to read a memory-mapped input from Basic you have to employ the USR function. This returns at the end of the machine code routine with a four-byte number made up of the contents of the accumulator, X and Y registers, and the processor status register in the least of most significant bytes in that order. Since the data is returned in the Y register we will have to AND the number with &FF0000 and then divide by &10000 (figure 1). To read the contents of a device at, for example, &FCFO, you would have to say:

```
A% = 146 : X% = &F0:
  value% = USR(&FFF4)
  PRINT (value% AND&FF0000)
  /&10000
```

	Most significant byte			Least significant byte
	Processor status register	Y register	X register	A register
AND with	& 00	FF	00	00
gives	& 00	Y register	00	00
DIV by		& 1	00	00
gives	& 00	00	00	Y register

Figure 1. Obtaining contents of Y register from a USR call

We now know how to read and write to devices on the 1 MHz bus, and next month I shall look at exactly how to connect things onto the bus, giving suggested circuits that you could use. This will be partly based on an applications note which Acorn produces called 'BBC Microcomputer Applications Note Number 1 – 1MHz Bus'. However, I suggest you do not write to them for a copy of this note as they have a minimum order charge of £10.

☐ Here is a PS for those computer buffs who already knew most of what I have been explaining. Can you find a neater way of picking off, say, the Y value, from the USR function? Using DIV &10000 MOD 256 seems as if it ought to work, but doesn't because the top bit of the processor status is set, which makes the number returned a negative number. If you have a simple solution send it to me via *Acorn User* at 53 Bedford Square, London WC1B 3DZ



Angered by the Australian Beebs

Sir, We ordered eight BBC Bs last February (the order got 'lost' twice) and eventually received them in October. Our intention has always been to network them using an Apple as file server and print server, and to use this network as a learning resource right across the curriculum. The arrival of the machines and, hopefully, associated software, has been eagerly awaited, by all concerned.

Imagine our disgust when (i) we hear from reliable sources that machines have been available in large quantities for a considerable time in Australia, (ii) when our machines do arrive they have MOS 0.1 and we are to be asked for a further £10 to bring the machines up to their original specification before we can even contemplate setting up our network.

We are more than happy with the machines and have offered to take part in software evaluation or pilot schemes, whether relating to Apple/BBC link-up or using the lab as a learning resource.

As soon as someone turns these machines into the machines we ordered there will be at least three networks in Britain.

Could you please put out a plea to anybody using Apples to get in touch for mutual assistance.

Can I end with a pat on the back for the magazine as a whole and a suggestion that you add a pupils' letters/problems page – given the number of Beebs in schools.

Mick Boston
Priory High School
Exeter

First things first. Any company which did not try to get its products overseas as soon as possible in this day and age would have to be barmy. And where does this 'large quantities' information come from? I doubt if the number in Australia is a large chunk of the 50,000 Acorn has so far delivered.

As for the 0.1 operating system, Acorn is sticking to its guns. Despite all the talk of taking the company to court, no-one has done so, yet.

Now, Apples as file servers. This isn't really on for a full Econet system. If you wish just to load and save programs, it should be possible to write the software, but it's not easy and there will be a lot of interface problems. However, we heard of a company that was trying to do this for a small network (Control Universal, Cambridge (0223) 358757). Also, RML 380Z's have been used (see last issue, p3). The MEP might be able to give you some contacts on this one.

So far letters from teachers, pupils and schools have been included with the other letters, as the points they make are usually relevant to everyone. If they do begin to deal with separate areas, then, who knows, another page may appear.

EPROM exchange

Sir, My model B has the 0.1 operating system in EPROM. How do I get it changed?

Also, I am having difficulty loading and saving programs. Will the dealer correct this free of charge?

I have read about the cartridge recording facility, but my keyboard has no 'hole'. Will the dealer alter the micro to allow for this?

P. Monckton

The EPROM will be exchanged for a ROM by your dealer free of charge. The service should be available now but check first.

If there is a fault in your machine, the dealer will fix it, but only as long as your guarantee is valid. Check all your leads first, and ensure you use the cassette filing system program in September's Acorn User which fixes a bug in MOS 0.1.

Cartridges are not available yet, but your dealer will sort out the hole when the time comes.

Z80 and Torch

First of all; BBC micro and *Acorn User*, both top quality.

What information is there on the second processors, Z80, 16032 and Gluon 32-bit, in terms of money and availability?

I would like to buy a double disc drive. Is it possible to use a 5¼" and an 8" floppy side by side, and if so can it be any 8" IBM compatible drive? Is the Torch Z80 the same as the Acorn Z80 second processor plus the BBC double drive?

Is it possible to download software through my colour TV which already has Teletext and Prestel?

To carry out a disk interface upgrade, I have to rely on dealers, but these are too far away. Can I send my micro to AB Electronics where it was assembled?

A lot of questions, but the level of expertise in your magazine led me to take the liberty of putting these forward.

Gert Jansz

The Z80 second processor should be available any time now for £195.50 and will support CP/M. The 16032 will be out within a year, but no further details are being released by Acorn, apart from those already given in Acorn User.

It is not possible to attach an 8" disc drive to the BBC micro at present because the intelligent software has not been written to cope with this data transfer rate.

As for Torch, their Z80 card is not the same as Acorn's, nor will the Torch disc pack the same.

Teletext and Prestel receivers are needed to download software, whether you have an ordinary TV, or one of the special ones.

The people you want for upgrades are Retail Control Systems, Gresham House, Feltham, Middx. Fitting a disc interface to a model B costs £132.25 including VAT and carriage.

You can run brands of discs other than the BBC's with your Beeb. However, because of the way the discs are formatted, it will be necessary to buy a utilities disc and manual from Acorn.

These are not supplied when the disc interface is fitted and normally come with the drives.

Cynics have claimed this is an attempt to stop people using other makes of drive, but Acorn have pointed out that the availability of the utilities disc means many makes of drive can be used.

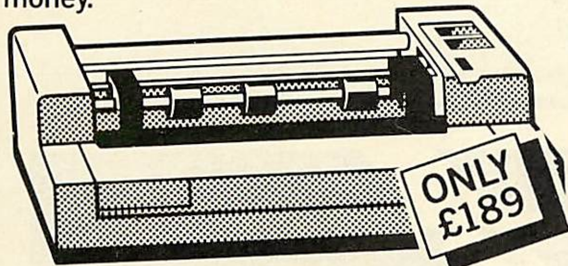
Disc plus manual will cost about £30, and will be available soon from dealers.

See Joe Telford's article in January's Acorn User for hints on using discs.

Write to: Letters, Acorn User, 53 Bedford Square, London, WC1B 3DZ

If you can buy these printer packages cheaper anywhere else MICROAGE ELECTRONICS Promise to give you the difference!!

At Microage you can now get the very best value in microcomputer printers. We've selected only the best printers available so you can choose confidently, knowing you're getting the best value for money.



Dust Covers Available at £3.95

ONLY £390

80 CPS

The Incredible Seikosha AP80

The AP80 is probably the world's LOWEST COST, compact 80-column graphic dot-matrix printer available.

It can produce single and double width characters and has the ability to produce any pattern through its dot-matrix capability.

The AP80's robust construction and its unique "unihammer" make it an extremely cost effective and efficient printer.

Its features include

80 Cols 30 cps ● Dot Matrix Unihammer action ● 96 ASCII standard characters ● Up to 8" paper width ● Upper and lower case ● Double width printing ● Standard interface: Centronics

The Amazing Seikosha AP100

Big brother to the AP80 the AP100 is a wider more sophisticated version of the AP80 with a full graphics capability and the ability to take standard width computer stationery, the AP100 is an ideal choice for anyone with a microcomputer.

Its features include:

80 cols 30 cps ● Dot Matrix Unihammer action ● 116 ASCII standard characters ● Full graphics capability ● Up to 10" paper width ● Upper and lower case ● Double width printing ● Centronics interface: ● Friction feed (optional extra). Only £215.

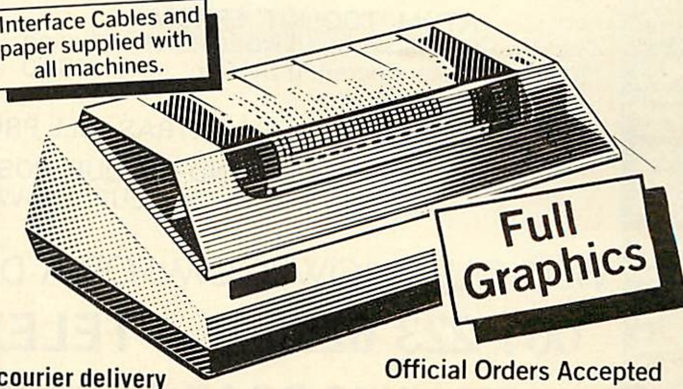
The Outstanding Epson MX80 FT Type 3

The MX80 is a flexible, high quality and extremely reliable graphic, dot matrix printer.

Its superb performance is due to a number of outstanding features. High resolution graphics, a wide range of type sizes, a quality 9 x 9 dot matrix head and up to 132 chars per line are just some of the exceptional features on this ideal printer for microcomputers. Only £390.

Specifications

132 cols 80 ● Logic-seeking, Bi-directional printing ● 9 x 9 Dot matrix printhead ● 96 ASCII standard chars with descenders ● Full graphics capability ● Several type sizes ● Automatic underlining ● Standard interface Centronics ●



Free courier delivery

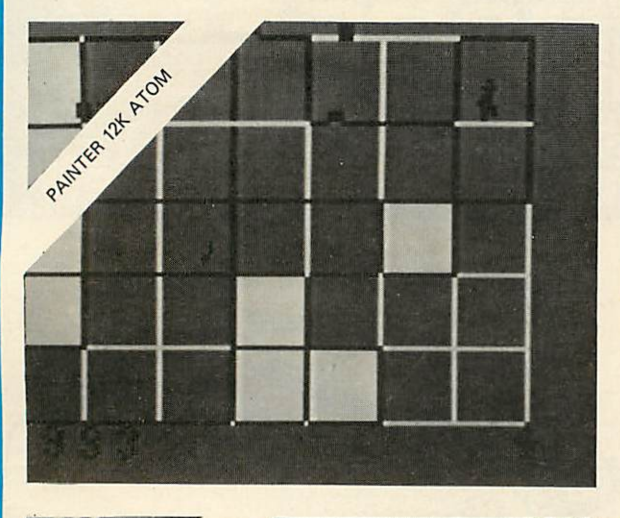
Official Orders Accepted

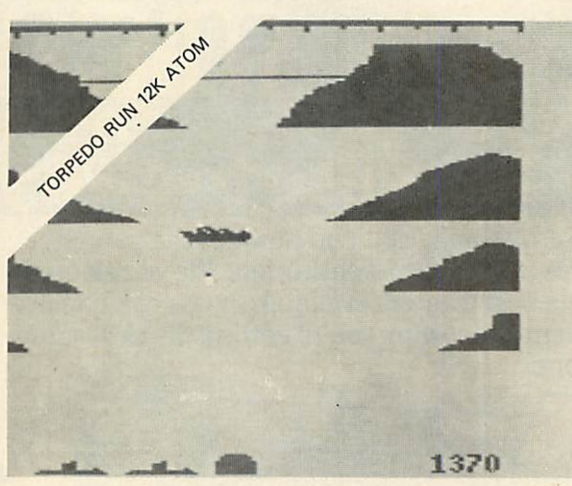
MICROAGE ELECTRONICS

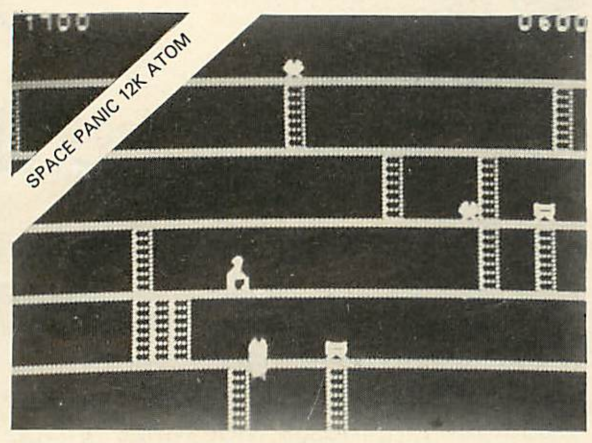
135 HALE LANE EDGWARE MIDDLESEX HA8 9QP TEL: 01-959 7119 TELEX 881 3241

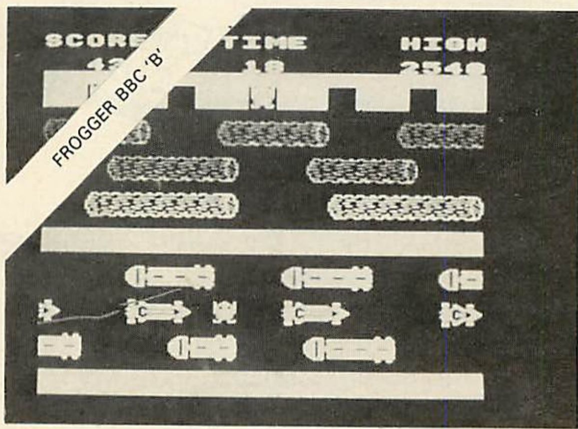
DEALER ENQUIRIES WELCOME CALL 01-959 7119 FOR TRADE PRICES

A+F SOFTWARE A+F









12K ATOM

Starburst	£5.75
Torpedo Run	£6.90
Cylon Attack	£6.90
Space Panic	£6.90
Painter	£6.90
Zodiac*	£6.90
Death Satellite*	£6.90

WE PAY 25% ROYALTIES FOR HIGH QUALITY PROGRAMS

Hours of enjoyment for all the family featuring fast moving graphics, sound effects and high score tables where appropriate.

BBC MODEL A

Tower of Alos*	£6.90

(above also runs on Model B)

BBC MODEL B

Lunar Lander	£6.90
Early Warning	£6.90
Road Runner	£6.90
Frogger (Machine Code)	£8.00

* Hours of Purgatory with these adventure games — can you get out alive?

BBC 'B' BRAND NEW INTERACTIVE ADVENTURE GAME WITH SUPERB GRAPHICS. 'PHAROAH'S TOMB' WILL BE THE BEST £8.00 YOU HAVE EVER SPENT ON AN ADVENTURE GAME.

ATOM 'TOOLKIT' EPROM

22 Extra Commands and Five features including 1200 Band Cass. operating System. £18.50

ADD 4 or 6 EPROMS TO YOUR ATOM WITH OUR 'ADDA' BOARDS

4-Way 'Adda' board	£20.75
6-Way 'Adda' board	£28.75

PLEASE NOTE: NO EXTRAS, ALL PRICES INCLUDE VAT AND POSTAGE

TO ORDER BY MAIL: SEND CHEQUE, POSTAL ORDER OR CREDIT CARD NUMBER OR TELEPHONE (24 HOUR ANSWERING) CREDIT CARD NUMBER

VISIT OUR SHOWROOM FOR A DEMONSTRATION OR MAIL ORDER

061-223 6206 TELEX: 667461 (Attn. A+F)

830 HYDE ROAD GORTON MANCHESTER M18 7JD

SINCLAIR SIRIUS SPECTRUM BBC DRAGON ATOM SINCLAIR SIRIUS SPECTRUM BBC DRAGON ATOM SINCLAIR SIRIUS SPECTRUM BBC DRAGON ATOM SINCLAIR SIRIUS SPECTRUM BBC DRAGON ATOM



'Whether your interests lie in business, educational, scientific, control or games applications, this system provides a possibility for expansion which is unparalleled in any other machine available at present,' comments Paul Beverley in the July 1982 edition of *Personal Computer World.*

The BBC Microcomputer can genuinely claim to satisfy the needs of novice and expert alike. It is a fast, powerful system generating high resolution colour graphics and which can synthesise music and speech. The keyboard uses a conventional layout and electric typewriter 'feel.'

You can connect directly* to cassette recorder, domestic television, video monitor. disc drives, printers (dot matrix and daisy wheel) and paddles. Interfaces include RS423, inter-operable with RS232C equipment, and Centronics. There is an 8-bit user port and 1MHz buffered extension bus for a direct link to Prestel and Teletext adaptors and many other expansion units. The Econet system allows numerous machines to share the use of expensive disc drives and printers.

BASIC is used, but plug-in ROM options will allow instant access to other high level languages (including Pascal, FORTH and LISP) and to word processing software.

A feature of the BBC Microcomputer which has attracted widespread interest is the Tube, a design registered by Acorn Computers. The Tube is unique to the BBC Microcomputer and greatly enhances the expandability of the system by providing, via a high speed data channel for the addition of a second processor. A 3MHz 6502 with 64K of RAM will double processing speed; a Z80 extension will make it fully CP/M** compatible.

The BBC Microcomputer is also at the heart of a massive computer education programme. The government has recommended it for use in both primary and secondary schools. The BBC Computer Literacy Project includes two series of television programmes on the use and applications of computers.

There are two versions of the computer. Model A, at £299, offers 16K of RAM and Model B at £399 has 32K of RAM.

For technical specification and order form, send stamped addressed envelope to P.O. Box 7, London W3 6XJ and for details of your nearest stockist ring 01-200 0200.

Broader horizons

BBC Model B Microcomputers are available for immediate despatch by courier to your door.

If you would like to take advantage of our special delivery offer just fill in this coupon, but remember this only applies to the Model B, and only in the UK. Any other items have to be ordered on the standard order form or from a BBC Computer Stockist.

BBC Microcomputer System Offer
c/o Vector Marketing
Dennington Estate
Wellingborough
Northamptonshire NN8 2RL

Please send me by courier ________ BBC Model B Microcomputer(s) at £399 including VAT and delivery.

Cheque enclosed for £ ____________ payable to BBC Microcomputer System readers a/c or charge
ACCESS BARCLAYCARD

Signed ____________
Name ____________
Address ____________

Postal Code ____________

Co Reg No 1403810 VAT No. 215400220

The BBC Microcomputer System

BBC Microcomputer System Offer, c/o Vector Marketing, Dennington Estate, Wellingborough, Northamptonshire NN8 2RL.

*Model A has a limited range of interfaces but can be upgraded to meet Model B specification.
**CP/M is a registered trade mark of Digital Research.

The BBC Microcomputer is designed, produced and distributed in the UK by Acorn Computers Limited.


PROGRAMPOWERMICROPOWER
GREAT BBC PROGRAMS FROM
BRITAIN'S LEADING SOFTWARE HOUSE
SWOOP
SWOOP (B) £6.95 — the NEW GALAXIANS
IT'S HERE AT LAST!! Galaxian-style, machine code arcade game. THIRTY screaming, homing, bomb-dropping, explosive egg-laying BIRDMEN, swooping down in ones and two's to destroy your laser bases. The exploding eggs feature makes a normally difficult game into a challenge 'par excellence.' Each new screen means increased difficulty. Bonus bases, score display, high-score and rankings are, of course included.
YOUR WAITING IS OVER!!
ALIEN DESTROYERS (B) £6.95
Sensational, high speed 'INVADERS' program with an abundance of features. Brilliant use of sound and graphics. 48 strong Alien Fleet of three different types plus Mothership scoring mystery bonus. Choice of six alien speeds and three bomb speeds. Vertical, angled and exploding missiles. Options to replace defences and suppress new fleet advances. Bonus bases awarded each new sheet. Scoring according to overall difficulty level. Ongoing display of score and hi-score. End of game rankings of top five scores.
This program has many unique extras e.g. 'battle analysis' showing the number of each alien type shot down, how many motherships destroyed, the number of sheets cleared, the shots fired, the percentage of hits made and the number of bases lost.
CHESS (B) £6.95
Our excellent machine code program — now with superb MODE 1, colour graphics. Six skill levels, play black or white, illegal moves rejected, 'en passant', castling, take-back of moves, and display of player's cumulative move-time. Options include Blitz Chess where you must move in 10 seconds, set-up of positions for analysis, replay of a game just played and saving of part completed games on tape. On loading, a 1972 Spassky/Fischer game can be replayed.
NOTE: Model A Version still available at only £4.95.
If you wish to upgrade your Model A version please return your tape, together with £2.50 plus V.A.T.
SWOOP
ALIEN DESTROYERS
CHESS
Other B.B.C. programs available:
Galactic Commander (B) £6.95/
Laser Command (B) £6.95/
Adventure £6.95/
Cowboy Shoot-Out (B) £5.95/
Filer £8.95/Micro Budget £6.95/
World Geography (B) £5.95.
Timetrek (B) £6.95/Spacemaze (B) £5.95/Martians (B) £5.95/Astro Navigator (B) £4.95/Star Trek £4.95/Maze Invaders (B) £4.95/Footer (B) £6.95/Munchyman £5.95/Seek £5.95/Eldorado Gold (B) £5.95/Cat & Mouse £4.95/Mastermind £3.95/Reversi 1 £4.95/Reversi 2 (B) £4.95/Roulette (B) £4.95/Gomoku £3.95/Zombies £3.95/Dissassembler £5.95/Constellation (B) £5.95/Where? (B) £5.95/Junior Maths Pack (B) £5.95.
WE Guarantee THAT ALL OUR ADVERTISED PROGRAMS HAVE BEEN COMPLETED AND ARE READILY AVAILABLE
WE ARE AUTHORISED DEALERS FOR ACORN ATOM, BBC MICRO & DRAGON 32
SPECIAL OFFER
Deduct £1 per cassette when ordering two or more.
MICRO POWER LTD.
8/8a REGENT STREET,
CHAPEL ALLERTON,
LEEDS LS7 4PE.
Tel: (0532) 683186
Please add 55p order P & P + VAT at 15%
VISA
Please Note:
All programs are now available at all good dealers or direct from MICRO POWER LTD.
WRITTEN ANY PROGRAMS!
WE PAY 20% ROYALTIES
FOR DRAGON, SPECTRUM
BBC, ATOM PROGRAMS